Product Engineering for silicon based MEMS IP

Vom Fachbereich Elektrotechnik und Informatik der Universität Siegen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Genehmigte Dissertation

von

Dipl.-Inf. Dirk Ortloff

1. Gutachter: Prof. Dr. Rainer Brück
2. Gutachter: Prof. Dr. Albert Albers
Vorsitzender: Prof. Dr. Hans Wojtkowiak

Tag der mündlichen Prüfung: 26.06.2006
urn:nbn:de:hbz:467-2287

Der Mensch wirft das Los;
aber es fällt
wie der Herr will.
Sprüche 16,33

Abstract

Today the silicon based Micro Electro-Mechanical Systems (MEMS) industry has inadequate development methodologies and frameworks. This is especially true when developing MEMS process and design Intellectual Property (IP). The existing development methods and frameworks are not accommodating for the necessary areas and components required for efficient, reproducible and controlled MEMS IP development. As such issues have not been addressed, this results in e.g. unresolved development methodology questions and non-systematic storage of product deliverables.

This thesis proposes a systematic development approach to tackle several of such insufficiencies. It proposes a novel MEMS IP development methodology guiding the developments from a product idea to a marketable product. Because the business processes of the method require tool support, a quality integrated Product Engineering Framework is proposed supporting the key areas. The thesis describes a new framework concept of business process components supporting the methodology and establishes processes guiding through the development efforts.

A major focus is set on the in-depth research of reproducibility and control aspects of the process IP development efforts. The findings concerning process development tracking are converted into novel concepts for supporting tools and the newly implemented software environment to address the found problem areas is introduced. The thesis presents results generated during the EU FP6 project PROMENADE [Eur04] and previous works. The software environment developed by the PROMENADE project builds a major cornerstone of a company wide framework for developing silicon based MEMS IP modules.

Zusammenfassung

Die Industrie für siliziumbasierte Mikrosysteme hat heute keine adäquaten Entwicklungsmethoden und -umgebungen. Dies gilt speziell für Firmen, die ein wissensbasiertes Geschäftsmodell (Intellectual Property, IP) verfolgen. Die existierenden Methoden und Umgebungen unterstützen nicht alle erforderlichen Bereiche und bieten nicht alle notwendigen Komponenten für die Mikrosystementwicklung. Diese unzureichende Unterstützung führt, unter anderem, zu unvollständigen und nicht reproduzierbaren Entwicklungsflüssen. Dies ist vor allem auf die unsystematische Speicherung der Entwicklungs- und Produktergebnissen zurückzuführen.

Diese Doktorarbeit stellt einen systematischen Entwicklungsansatz vor, der mehrere dieser Probleme bearbeitet. Hierzu werden alle relevanten Aspekte analysiert und exemplarische Lösungsvorschläge aufgezeigt. Es wird eine neue Methodik bereitgestellt, welche von der Idee bis zum vermarktbaren, wissensbasierten Mikrosystemprodukt führt. Hierzu wird eine Umgebung aus Geschäftsprozessen und Unterstützungswerkzeugen für die Methodik beschrieben.

Ein besonderes Augenmerk wird auf die detaillierte Analyse von Reproduzierbarkeits- und Kontrollaspekten der Fertigungsprozessentwicklung gelegt. Die Ergebnisse wurden zur Entwicklung neuer Konzepte und Werkzeuge zur Entwicklungsunterstützung in der Praxis eingesetzt. Die Arbeit präsentiert Ergebnisse aus dem von der EU geförderten Projekt PROMENADE [Eur04] und des Vorläuferprototypen PDTS. Die innerhalb des Projektes entwickelte Softwareumgebung bildet einen Grundpfeiler für die firmenweite eingesetzte Entwicklungsumgebung.

Contents

Glossary

Fabless If a company develops a semiconductor product without owning a semiconductor fabrication line, the products are developed fabless.

IDDQ IDDQ is a test technique based on measuring the Quiescent supply current of a CMOS transistor.

IP Intellectual Property. The legal rights associated with inventions, artistic expressions and other products of the imagination, in the form of patents, trademarks, service marks, trade names, trade secrets, and copyrights.

Operations The operations efforts of a business valuate products, services and assets marketed by a business. Therefore operations takes the results of a development project through the life cycle of the product. It can be distinguished from a project because it is ongoing and repetitive.

Product As [Wik05i] defines, a Product is anything that can be offered to a market that might satisfy a want or need of a potential customer. It is the complete bundle of benefits or satisfactions that buyers perceive they will obtain if they purchase the product. It is the sum of all physical, psychological, symbolic and service attributes.

Product Engineering (PE) Product Engineering is the discipline of driving and managing product development efforts following a pre-defined methodology. It considers the whole life-cycle of a product from its idea over the various development stages to a marketable product. A more detailed definition is given in section 2.1.

Product Engineering Framework (PEF) The discipline Product Engineering requires tool and process components to perform the related tasks. The Product Engineering Framework contains such a tool framework supporting the efforts to drive a product idea to a marketable product. For the definition of the different steps it uses a development methodology defined in the Product Engineering Method(PEM) integrated with the Quality Assurance Framework.

Product Engineering Method (PEM) Defines a methodology consisting mainly of processes and procedures along which product development efforts are driven forward. The components of the methodology consist of business processes and sub-processes defining the sequence of actions to develop a product idea into a marketable product.

Program A program is a collection of projects or product developments to address a strategic objective of the overall strategy of a business.

Project A project is a temporary endeavour to create a product, a service or any other result like a cultural change. Therefore a project ends when its objectives are achieved, e.g. a product is launched. It is typically constrained by scope goals, time goals and cost goals. For product development all efforts to evaluate a product idea and develop a product are considered to be a development project. Therefore a project is the development phase (from the idea to a marketable product) of the product life cycle and hands over the development results to operations (defined above).

Project Management (PM) are the activities to guide a project via the application of knowledge, skills, techniques and tools typically performed by the Project Manager.

Quality Assurance Framework (QAF) The Quality Assurance Framework defines necessary quality assurance activities. It comprises of e.g. review procedures and quality audit procedures ensuring that product deliverables are developed until a certain quality level is reached and that they are developed in accordance to a predefined quality assurance model.

Abbreviations

Abbreviation	Explanation
CM	Configuration Management
CMMI	Capability Maturity Model Integrated
CMS	Content Management System
COO	Chief Operations Officer
CORBA	Common Object Request Broker Architecture
CR	Change request
CTO	Chief Technology Officer
CVS	Concurrent Versions System
DFM	Design For Manufacturability
DoE	Design of Experiments
EDA	Electronic Design Automation
FMEA	Failure Mode Effect Analysis
GUI	Graphical User Interface
HDL	Hardware Description Language
IDL	CORBAs Interface Definition Language
IIOP	Internet Inter-ORB Protocol
IP	Intellectual Property; described in section 2.2.1
IPPD	Integrated Product and Process Development
ITIL	IT Infrastructure Library
JAR	Java ARchive
J2EE	Java 2 Enterprise Edition
MEMS	Micro Electro-Mechanical Systems
MES	Manufacturing Execution System
M&S	Marketing & Sales
OGC	Office of Government Commerce
OMG	Object Management Group
ORB	Object Request Broker
PE	Product Engineering
PEF	Product Engineering Framework
PEM	Product Engineering Method
PM	Project Management
PMBOK	Project Management Body Of Knowledge
PRINCE2	PRojects IN Controlled Environments 2

Abbreviation	**Explanation**
QA	Quality Assurance
QAF	Quality Assurance Framework
R&D	Research and Development
RE	Requirements Engineering
RMI	Remote Method Invocation
RS	Requirement Specification
RTL	Register Transfer Level
VHDL	Very High Speed Integrated Circuit Hardware Description Language
XP	eXtreme Programming

1. Introduction

Unlike in microelectronics the MEMS domain is characterized by a variety of different devices fabricated in almost arbitrary shapes using all kinds of geometries, materials, process steps and so on. On top of that the MEMS process and design IP development is much less standardized. While the Intellectual Property (IP) business model is relatively common in the microelectronics area, only a few companies apply this model in the MEMS industry. This IP business model is based on the strategy to develop and protect a piece of knowledge (IP) e.g. a microelectronic circuit design or a manufacturing process sequence. If the developing company does not own their own semiconductor fabrication line, the products are developed fabless. The developed knowledge pieces are then licensed to chip design houses or semiconductor manufactures to be used in their products or offered as part of their own process option portfolio (see more details in section 2.2.1). For this MEMS IP industry up to now hardly any appropriate methodologies or frameworks for describing, performing and tracking the development efforts exist. Introductions into the MEMS field, the essential differences to microelectronic design, its properties and the necessary design support environments have been recently published in [Pop05] and [Wag05] and should not be repeated here.

Another distinguishing factor between the microelectronics and MEMS domain is the more diversified knowledge required for successful MEMS development. MEMS developments require a multidisciplinary team structure. The professions, mindsets and backgrounds of the members of such teams tend to be more differentiated than in the pure VLSI industry. By that the required product development method and the supporting engineering framework needs to be more detailed, stricter and needs to cover more aspects. Similar social implications for the software industry have been described in [Cop94] [Coc96]. With respect to the EDA tools embedded into such an environment this will require new methods and tool concepts and potentially the integration of more diverse tools.

This thesis proposes a systematic development approach to tackle several of such insufficiencies in current research and development practices. It proposes a novel, generally applicable and quality integrated MEMS IP development methodology. The methodology is based on industry accepted standards tailored to the specific needs of the addressed industry. For the creation of an efficient, reproducible and controlled framework for developing MEMS IP, several areas require consideration. The thesis provides insight into the identified areas. It introduces a new MEMS Product Engineering approach, a new framework of business process components and processes guiding through the development efforts. Additionally a newly assembled set of tools supporting the processes is described. Therefore it introduces a new method and framework to engineer MEMS process and design IP products.

The development methodology is based on the generally accepted idea of treating the development efforts as a process. This process uses several steps and iterations while each step ends in a quality gate. During these gates, management takes a well considered Go/Hold/Recycle/Kill decision about the development project. The quality aspects are built into the gates after each development step. Additional quality assurance activities run in parallel to the efforts supporting the development project. This ensures successful delivery of controlled and high quality results. The overall progression is managed by a project management framework based on the well established project management method PRINCE2. This method has been tailored to the specific needs of the industry. A brief outline of the quality integration and the different development steps is presented in figure 1.1.

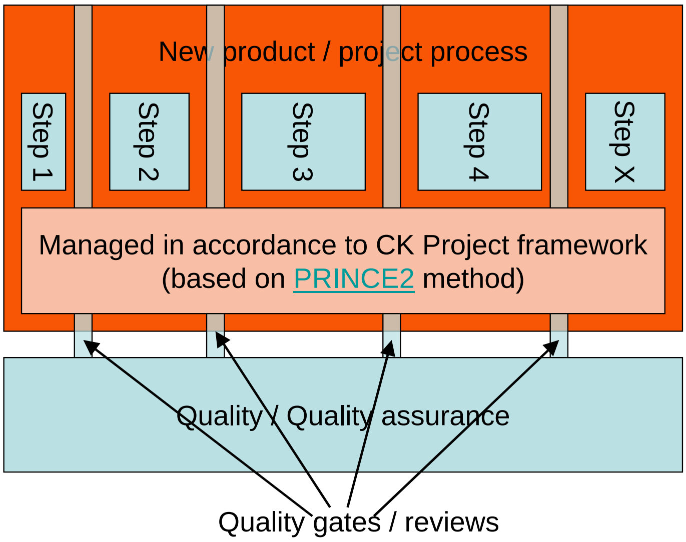

New product / project process
Step 1
Step 2
Step 3
Step 4
Step X
Managed in accordance to CK Project framework (based on PRINCE2 method)
Quality / Quality assurance
Quality gates / reviews

Figure 1.1.: Integration of the product development process and quality

The above outlined methodology and its business processes require a certain set of supporting tools to ease and speed up the efforts. The requirements for theses tools have been analysed and different areas of concern have been identified. The needs will be described and grouped into different building blocks so that they can be supported by existing software tools of different categories. Many necessary aspects can be covered by a careful selection of existing software components but for other aspects new tools have been developed. These especially provide necessary integration or cover aspects where no existing solution could be found or existing solutions were not suitable or feasible. The tools of the different blocks are taking all the different necessary aspects into account. They build a complete Product Engineering Framework of components supporting the development efforts. The selection and integration of the tools is unique in its kind, because it specifically supports the very young industry of MEMS IP developments.

A major focus is set on the in-depth research of the essential knowledge management for process IP development. Especially reproducibility aspects of these development efforts are considered and the development tracking and issues resulting from this area are assessed. During these activities a systematic analysis of the technology and structure driven design was performed. It turned out that a lot of unstructured data is accumulated and non-systematically stored. Also most of the collected data is more or less related and these relations are usually not properly archived. Therefore the organized management of the data is a key topic in the design of microsystems, especially for the development of manufacturing processes. Because current categorization and archiving techniques are not sufficient, a new storage and retrieval concept for this type of data and knowledge has been developed. The basic concept is generic for many industries e.g. semiconductor industry, pharmaceutical development industry. It is applicable in those industries where the development uses a hierarchy of development items. It applies where one item consists of a set of subitems, generated and used during an experimental verification process. This thesis presents the new concept, describes insights gained and results generated during the efforts of a feasibility study applying the concept to the silicon based MEMS IP development industry. The developments of a prototype software environment called PDTS (Process Development Tracking System) will be described, and the results have been published in [OVC05].

The gained insights and concepts of this proof of concept have been used in the EU FP6 project PROMENADE [Eur04] and have been re-implemented and extended during that project. This extended system, called EVERIST, has been integrated with the Process Design Environment of the project, described in detail in [Pop05] and [Wag05]. The intermediate and final results of the PROMENADE project are reported in e.g. [WPS+05], [WSP+06] and [SOH+06]. They were used as one major cornerstone for the overall efforts of generating a company wide framework for developing silicon based MEMS IP modules. Focal points of the whole developments are knowledge bases for different purposes structuring and enabling the fabless IP business model. The basic systems use relational database schemes to store all collected data and knowledge and enable to relate it in a useful manner.

The thesis presents an environment of software systems, business processes and methods. The Product Engineering Framework and the integrated quality framework put the methods, building blocks, tools and procedures into practical use. The engineering framework uses modern technologies like Java, J2EE, Intranet as well as standard components like Revision Control Systems, Content Management Systems and so forth. The environment enables collaborative work between several sites and users. They may use different hardware and operating system platforms and the environment specifically supports the fabless MEMS IP business model.

The remainder of the document has the following structure. The second chapter describes the State-of-the-Art and the identified requirements for MEMS IP product engineering. Taking these descriptions as the motivation, the third chapter describes existing product development and project management methods and approaches. From that it derives the newly developed product engineering method specifically suited to the fabless silicon based MEMS IP business model. The following chapter describes the requirements for and the newly assembled product engineering framework. It supports the processes of the PE method providing essential building blocks for a working environment. This chapter is followed by chapter 5 briefly describing existing and incorporated quality and quality assurance principles. The newly developed environment supporting the MEMS process development is described in detail in chapter 6. This environment supports the process design and verification efforts following a three step verification approach from formal process consistency checking to support of experimental verification. The later is supported by the novel approach allowing for result capturing and retrieval. The intended further work is summarized in chapter 7 and the conclusions in chapter 8 finalize this thesis.

2. Status and Requirements of MEMS IP Product Engineering

2.1. Product Engineering introduction

Before discussing Product Engineering in detail, clarification is required as to what this term means. Besides the diverse basic terms defined in the Glossary, the definitions of Product Engineering for different disciplines (manufacturing, software development, etc.) reported in literature ([Ree05], [Bux05] and [NIS05]) vary slightly. But the fundamentals in these definitions are the same. They consider Product Engineering to be the role and task to drive product developments from the product idea through the whole product life-cycle (design, development, manufacture, usage and recycle). Another typical characteristics of Product Engineering is that it needs to include the integration of multi-disciplinary teams. The distinguishing factor from ”normal” project management is, that it even has to consider the time after usage. Project management typically only guides the development phase. This is due to the nature of projects being a temporary endeavour, as defined in e.g. [PMI05]. This implies that they do not last as long as a product life cycle. The importance of Product Engineering itself is underlined by a program set up by the National Institute of Standards and Technology (NIST) in the USA described in [NIS05]. This program aims to define an exchange format for product representations of CAD solutions for the different engineering disciplines and improve the interoperability between the system of different disciplines. This is necessary because product designs become increasingly multi-disciplinary.

By definition the task of Product Engineering is to guide the new product development efforts. To understand the boundary constraints of new products and to get a clearer picture, it needs to be defined what characterizes the development of a new product. Drivers for innovation and for new products are technology advances, changing customer needs, etc. There exist several categories of newness of a product e.g. new to the world, new product line for the company and so forth. These categories have a high impact on the business case of a potential new product and need to be considered when taking a development decision. These aspects and a more detailed categorization of new products will be given in section 3.1.

Another area to be investigated are the key ingredients for successful new product developments. As documented by Cooper et.al. in [Coo98], their analysis revealed that there are several key ingredients e.g.:

- To embrace a long-term commitment to new product developments,
- to develop a vision, objectives and strategy for new product development efforts driven by and linked to the business's corporate objectives and strategy,
- to install a systematic, high-quality new product process.

Several of the success factors are non-systematic and depend on company politics and management decisions. These non-systematic aspects should not be further investigated in the current context. The aspect to have a high quality new product process or product development methodology will be discussed further in chapter 3. It introduces a newly developed product engineering method specifically targeted for silicon based MEMS IP developments.

The next question to answers is about the necessary ingredients for a high quality new product process. The first major ingredient is to have a methodology to coordinate the life cycle of a product. This mostly consists of processes for the development, the improvements and the phase-out of a product. Because a pure methodology only sets the processes how to perform the tasks, the second necessary ingredient consists of a framework of tools supporting the different activities. Into both ingredients the quality aspects needs to be build in to achieve a high quality product. The following sections describe the State-of-the-Art in these necessary fields and present the findings of improvement potentials. Section 2.4 will summarise the issues addressed by this thesis.

2.1.1. Implications of Product Engineering

Setting up product engineering activities can have several implications onto a business which are e.g.

- Implications on Business processes
  Existing business processes/practices/procedures might be formalized introducing diverse changes. New business processes will be created. All processes should be as lightweight as possible and should always consider the specifics of the individual project in terms of size, risk and so forth. These generic product engineering processes can and need to be adapted to the individual project requirements but this adaption needs to stay within the defined boundaries.

- Additional business groups
  Potentially additional business groups will be created, if not already established. Definitely there has to be a separate quality group, responsible for all quality and quality assurance related activities. This group needs to be independent from the other business groups to allow effectiveness. Additionally there should be a product engineering group with the role defined in section 2.1.2.

2.1.2. Product Engineering role

The role of Product Engineering and the Product Engineer(s) inside a cooperation can be summarised as:

- Being the 'Missing Link' between Research&Development and Product Management, Program Management or Marketing, depending on the setup

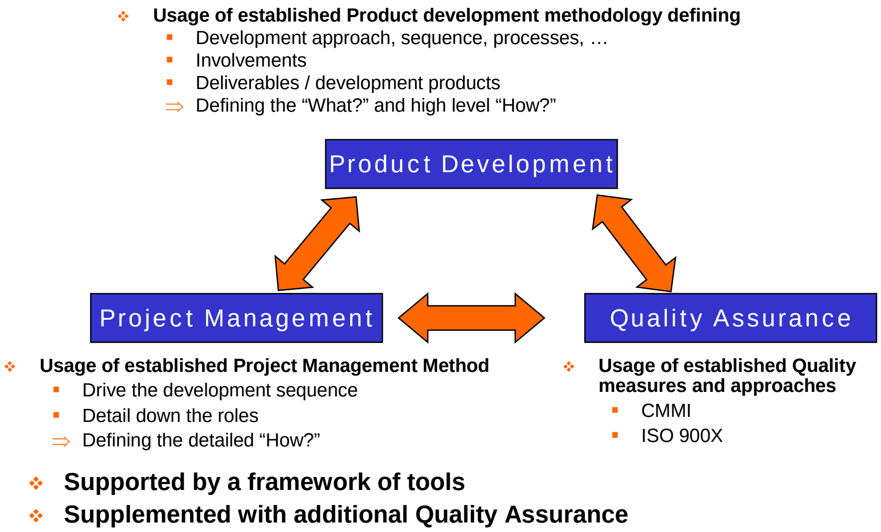
- Mediating between Marketing/Product Management and the development organization groups (e.g. IC design, process development, device modelling) having the tasks of

  - Pulling the strands together,
  - driving the product development process from the product idea to a marketable product ,
  - develop and maintain
    - the Product Engineering Method(PEM),
    - the Product Engineering Framework (PEF) and
    - the Quality Assurance Framework(QAF), together with the quality group
  - For bigger projects or product developments the team leader of a development project might be provided by a PE group. Another scenario is that the Product Engineer only supports the program or project managers which are responsible to manage the developments (matrix org.).

The three areas (PEM, PEF, and QAF) are closely related as shown in figure 2.1. They need to be successfully integrated for developing products. The outline of these interactions is presented in figure 1.1 and will be detailed further in the following chapters.

2.2. State-of-the-Art

The abstract and introduction briefly introduced the motivation for this work. The current chapter goes into more detail about the State-of-the-Art in MEMS



- **Usage of established Product development methodology defining**
  - Development approach, sequence, processes, …
  - Involvements
  - Deliverables / development products
  - ⇒ Defining the "What?" and high level "How?"

Product Development

Project Management

Quality Assurance

- **Usage of established Project Management Method**
  - Drive the development sequence
  - Detail down the roles
  - ⇒ Defining the detailed "How?"

- **Usage of established Quality measures and approaches**
  - CMMI
  - ISO 900X

- **Supported by a framework of tools**
- **Supplemented with additional Quality Assurance**

Figure 2.1.: Interactions between the different areas to take into account

IP Product Engineering and the required development support areas. Therefore this chapter will examine the development methodologies and practices, support tool components, the existing solutions and it will describe the encountered lacks. Additionally it will give an evaluation of the current approaches, methods and tools.

Of special interest is the State-of-the-Art in the area of MEMS process development and the support tools for such ventures. Here the focus is especially on tools enabling systematic knowledge acquisition and retrieval. This is required to make the development efforts reproducible, transparent, controlled and allow to generate IP knowledge packages, marketable as a product.

2.2.1. IP Business model

In essence the IP business model consists of the strategy to generate knowledge in the form of Intellectual Property and license/sell this knowledge as packages to other companies. In terms of microelectronics design this would imply to generate a chip or cell design and license the usage of this design to another company. An IP company typically does not intent to manufacture or use this knowledge for own physical products, although some companies use a mixed business model. After licensing, the customers of an IP product are allowed to use this knowledge inside their own products. For electronic designs this could denote that the customer uses an acquired cell design in their own chip design. For design IP different types like Soft-IP, Hard-IP, Processor-IP can be distinguished. Additionally the providers use different IP product business models e.g. per-use, time-based or royalty-based. A very good description of the different types, business models, pros and cons of the different approaches and what providers and customers have to take into account when considering acquiring design IP is described in [Fab05]. Additionally that document gives quite some more background information about the matters of the design IP business model. It even gives references to exchange and information platforms enabling design houses to find appropriate semiconductor design IP for reuse. One example for such platforms is [Des05]. The IP business model can even be taken a step further to combine different designs into Virtual Components (ViC) as described in [Dol05].

For process IP, similar approaches like in [Fab05] can be used. Only the category of potential customers is different and the process IP segment of the industry is even more immature than the design IP industry. The above cited document describes the semiconductor design industry and targets the pure chip design efforts. Semiconductor/MEMS process IP will be primarily used and bought by semiconductor manufactures. They can integrate the acquired manufacturing (sub-)process as a front-end or back-end option into their portfolio. As an example, they could offer a non-volatile memory option in their back-end processing as a standard process option. The manufactures then are able to offer this new option to their own customers (internal or external design groups). Process IP is typically offered together with a single device or block design to use the manufacturing option. Full custom cell designs, e.g. a non-volatile memory block, can be offered as an additional option. This would follow the above described design IP business models and could be offered either by the foundry or by the provider of the process IP.

2.2.2. Process IP transfer mechanisms

The key enabler for the application of an IP business model are the mechanisms to transfer the knowledge from the vendor to the customer. For design IP this transfer is more or less a standardized procedure. It contains handing over layout files for the different items, simulation models (potentially for several different purposes), simulation and test results like DRC, LVS logs etc. and documentation. A standardization initiative addressing this area has been formed by several market leaders. They have formed the SPIRIT consortium [SPI05] addressing these issues. An example for a standardized IP exchange platform is ”The IP Portal” [VCX05]. Anyhow a generally accepted mechanism and bundling scheme is not defined yet, even for the design IP industry. A similar initiative and a kind of ”quasi-standard” is necessary for the process IP as well. Additionally tools for transferring the process IP packages would be beneficial to enable the customers to compare own processing results with the baseline results of the process IP package. Evaluating the market, currently no sufficient electronic means for the process IP bundling and transfer could be found.

2.2.3. Product Engineering Education

Unlike in other industries, for the MEMS industry at present no product engineering methods or education program could be found in literature. For other industries, like the Semiconductor Industry, lectures and education lines about product engineering in that industry can be found. One example is the PSPE program of the Texas Tech University which is described in more detail on [Tex05]. Another example is the ”Product Engineering and Manufacturing” program for the mechanical industry based on a textbook [COL05] and being taught at the College of Engineering at the University of Maryland. More examples of education for the Product Engineering disciplines in different industries are the PACE education project [Par05] for the automotive industry, the EFCE Bologna Recommendations on education documented in [Eur05] for the chemical engineering or the ECSEL initiative, documented in [ASE95], for mechanical engineering. For the MEMS industry such programs do not yet exist although first steps are initiated by the Micro and Nanotechnology Commercialization and Education Foundation (MANCEF) [Mic05a] but have not resulted in formal education programs yet. Recently first activities into this direction have been started by the Southwest Center for Microsystems Education. They have defined an industry survey([Sou06]) to acquire input for the development of appropriate education courses. Additionally they aim to define a set of skill standards and competencies required for MEMS and Product Engineers.

2.2.4. Product Engineering Methods

In the field of Product Engineering methods for MEMS a similar situation as in the education field has been detected. The education programs of the other industries listed above introduce methods to engineer a product from its idea to a marketable product. An example method and guidelines for the software industry are given in [Sof05b]. These have been formalized into the overall method and maturity model of the Capability Maturity Model (CMM), documented in e.g. [Sof05a] and [Wik05a]. This model has been extended with several additional aspects from other industries and is now known as the Capability Maturity Model Integrated (CMMI), documented on the CMMI homepage [Car05] and in [CKS03]. These models describe how to engineer products in a controlled manner and define criteria to evaluate and define the maturity of an organizations development procedures. Further examples are the method blue-prints Stage-Gate™ and IPPD (Integrated Product and Process Development). Stage-Gate™ belongs to a family of generic methods using a linear development process consisting of development stages and evaluation gates. IPPD belongs to the family of iterative, information and decision based methods. While the linear staged methods have a relatively fixed set of stages executed once where each stage can itself be iterative, the iterative methods cycle the whole development loops several times. Both method families are described in more detail in section 3.2.

For the MEMS and semiconductor industry a model and methodology suited to the industry could not been found in literature. Although the first MEMS devices were developed in the 1967 ([Nat67]), the commercial market of MEMS based products is relatively young. First commercially used MEMS devices appeared in the beginning of the 1990's, as reported in [All05]. The MEMS IP market is even younger. Therefore there are only rare examples of companies selling MEMS process or design IP modules. This is, at least partly, due to the fact that there is no abstraction like design rules in standard VLSI electronics design for MEMS device designs. Such an abstraction could decouple the device design from the manufacturing aspects. Due to the multi-domain (electrical, mechanical, etc.) nature of MEMS devices and the usage of the third dimension for these devices, the influences of the manufacturing process onto the structural and behavioral device design stages are much more severe. For MEMS an Application Specific Process Design (ASPD) is typically necessary, as reported in [WPH03] and [PWOB04], requiring a special manufacturing process for every device. This industry status might cause the lack of engineering methods for MEMS IP products. Anyhow more and more companies consider a fabless and IP based business model for MEMS, applying a similar business style like in the VLSI electronics industry. An overview how a fabless model can be applied for the development of MEMS-enabled Products is described in [MRI05].

So there is a need for an engineering method suited to develop MEMS IP packages which currently is not reported in literature. Therefore this thesis will address the need and introduce a development method for such products. This new method uses the blue-print product development method Stage-Gate™ as a baseline. Stage-Gate™ is used in various industries and can be considered as one of the two market leaders. This product development method is combined with the project management method PRINCE2. The combination of the two, together with a new sub-method for the process development, is tailored to the specific needs of this industry and is introduced in chapter 3.4.

2.2.5. Product Engineering Frameworks

As a consequence of missing product engineering methods in the MEMS industry, the software market does not provide tools or frameworks for the complete set of tasks of MEMS product engineering. Another reason is the size of the market, although it is constantly growing with a high rate. Such tools and frameworks are required to ease the collaboration of this multidisciplinary, challenging industry. As outlined in [All05] this is currently an open issue.

For other industries such tools and frameworks like Teamcenter Engineering of UGS ([UGS05]) and CATIA V5 from IBM ([IBM05a]) exist.

For the MEMS industry, apart from part solution like CoventorWare from Coventor [Cov05] or IntelliSuite 8 from IntelliSense [Int05a], no complete engineering framework is available. CoventorWare and IntelliSuite8 cover only parts of the necessary tools not supporting all necessary areas to engineering a MEMS product. Although in [SGCD02] the authors from Coventor claim to have a full product engineering solution, their scenario only supports single or a small number of devices and does not support detailed process simulations. Additionally product life-cycle aspects are not covered because the tools basically only support the development process and even there only in parts. Additionally these tools are not really suitable to develop IP products because they are more manufacturing oriented and do not support the export into IP packages. The behavioral modelling and analysis of devices is an important part but for the complete process IP development, the aspects of e.g. support for experimental verification is missing.

2.2.6. MEMS process development

The development sequence of silicon based microsystems is currently divided into two major subtasks:

- The development of the fabrication processes determines the specific characteristics with regard to the material and the vertical properties of the devices.
- The design of the structural aspects determines the lateral properties often described in layout data in analogy to VLSI design.

Both subtasks have a high degree of interdependencies and must be performed concurrently. This is obvious as far as MEMS, MOEMS etc. are taken into account but becomes more and more important also for microelectronics. Their shrinking dimensions and processes are driven to the technological edge which results in more and more technologically motivated properties that have to be taken into account during the circuit design phase. This implies that even for the circuit design, the earlier described abstraction using design rules and their verification becomes less reliable. In the moment lateral design of micro structures is supported by tools either coming from microelectronic design (e.g. frameworks like Cadence, Mentor etc.) or from mechanical design (e.g. AutoCAD etc.) without concerning manufacturing process design. Even tools and frameworks specifically targeted towards MEMS design like CoventorWare from Coventor [Cov05] or IntelliSuite 8 from IntelliSense [Int05a] are based on microelectronics tools and design flows. They do not really cope with the requirements of application specific processes because they primarily are focussed on the behavioral aspects of the devices.

On the other edge there are two main commercial vendors providing tools for two and three dimensional numerical simulation of semiconductor processes (TCAD), namely AVANT! (now acquired by Synopsys [Inc05]) with the tools TSUPREM-TAURUS and DIOS and SILVACO [SIL05b] with the tools ATHENA, Clever and Victory. All these tools provide several empirically or physically based models for the essential process steps but the major focus of all these tools is CMOS technology. Nevertheless, although similar processing steps are applied to silicon based MEMS processing, some empirical models useful for CMOS are inappropriate for specific MEMS processes. The main difference between the MEMS and the CMOS area is that genuine CMOS focuses more on the electrical properties of the resulting layers while MEMS design focuses more on their mechanical properties.

Currently, silicon based MEMS process development is often based solely on the design of experiments, execution, evaluation and qualification within a semiconductor manufacturing and development facility. Development proceeds in cycles of generation and validation, until the objectives of the development (step) have been achieved. This requires a significant investment in engineering, fabrication, characterisation and qualification resources. On top of that, due to the lack of a proper starting point for the development, convergence towards an acceptable solution is not even guaranteed. This is due to the enormous solution space which has to be handled. This can lead to a significant waste of time and the abandoning of possibly successful devices due to the inability to design and implement a proper manufacturing process. Additionally the integration of the device and circuit design activities with the efforts of the process development activities is limited.

Also lacking, is the area of knowledge management with respect to the process design and management. Today technology data derived from simulations, tests or measurements is usually kept informally and non-systematically on paper, in spreadsheets or merely in the minds of process engineers. Hence it is hardly accessible for use in future process development projects. Tools like Promis [Bro05] marketed by Brooks Automation or Workstream [App05b] and FAB300 [App05a] from Applied Materials are more focused on issues related to the management of production lines. They are mostly concerned with scheduling, throughput, production control, yield improvements, and are not targeted for process development. All these tools belong into the Manufacturing Execution System (MES) category and are basically suited for production rather than development purposes. These tools are powerful, very complex and expensive and therefore less suited for the fabless IP business model. Additionally those tools miss features incorporating measurements results (pictures, material composition analysis, etc.) into the knowledge base relating them to the runs. Especially they do not perform sufficient document management of development artefacts. The same is the case for the products like

FlowDB and OperatorDB from PhoenixBV [Pho05]. These products are using a concept to track the development sequence similar to MES systems and try to link those into the development/process and device design. The drawback here is as well that they miss out the incorporation of the knowledge resulting from all kinds of artefacts (reports, pictures, analysis files, etc.) and that they do not mesh this data into the overall picture and do not relate them to the runs.

As a result of this disregard of technological degrees of freedom within the design flows, process flows and knowledge management, the MEMS community suffers from diverse uncertainties concerning the fabrication process and its effects on developed MEMS devices. These insufficiencies imply limited reproducibility, less knowledge gain and unnecessary development resource usage.

2.3. Requirements of MEMS IP Product Engineering

2.3.1. Process IP transfer mechanisms

As described in section 2.2.2, currently no standards and no support environment for the transfer of process knowledge are reported in literature. Therefore there is a need to establish standards and tools providing export functionalities as an exchange method for process IP packages. These tools and environments need to allow to carefully collect the knowledge and track the development efforts with all results. Additionally they need to be able to separate out certain knowledge packages and need to support the electronic export and transfer of these IP packages.

Therefore a new approach to address these transfer issues is required. This thesis addresses this by proposing a new approach which has been integrated into the PROMENADE project [Eur04]. One focus of the PROMENADE project is the process IP protection and distribution. The components of the environment use a strict user and role management to protect all parts of the process knowledge. The systems selective and rights maintaining export and import functionalities allow the bundling and protection of IP packages. It allows to export and import process recipes together with simulation and experimental verification results captured in the different components of the system. More details about the approach and the possibilities by this newly developed system are provided in section 6.5.2.

2.3.2. Development Methodologies

As described in section 2.2.4, currently no MEMS IP engineering methodology could be found in literature. Additionally no education programs for this area are know. Therefore it is required to develop and introduce a specific method to engineer MEMS IP products. The new method introduced by this thesis and described in section 3.4, is based on the Stage-Gate™ product development method and the PRINCE2 project management method. Stage-Gate™ is widely used throughout diverse industries and PRINCE2 has become the European Standard for project management. Because the Stage-Gate™ method focusses on the ”What” and only indicates the ”How” on an abstract level, the PRINCE2 method has been used to supplement the basic product development method. Both methods together build a good starting point to develop a new method targeted to allow product engineering for the silicon based MEMS IP industry. Into this method a novel process development approach has been integrated. Because the process development activities and the process IP is the major part of the intended business model and no structured approach existed, it became necessary to have a special eye on supporting these efforts. The innovative process development method introduced in section 3.5 structures the process development approach and is supported by a software environment introduced in chapter 6.

2.3.3. Engineering Frameworks

A similar situation as for the product engineering methodologies holds true for the Product Engineering Frameworks. Because the MEMS IP industry is rather new and very multi-disciplinary, no Engineering framework solution directly supporting all needs could be found on the marketplace. Therefore the development of an integrated framework covering all necessary aspects for MEMS IP development is desirable. The unique framework introduced in chapter 4 addresses this issue in proposing a suitable solution assembling and integrating available software solutions with additional integration components. The overall framework allows to engineer a MEMS IP product from the product idea to its market launch integrating and providing tools for all necessary disciplines. A major cornerstone of the defined and implemented Engineering Framework is the process development environment introduced in chapter 6.

2.3.4. Process development Support

As described in section 2.2.6, all tools on the market place offer only partly the functionality required for MEMS physical design stages. A market research has established that no tool framework of any vendor and no combination of tools from different vendors serve the need of a continuous design flow taking structural and process demands into account. For current and future applications it seems unavoidable to take into account process design concurrently performed with structural design. Only the concurrent approach enables results with predictable structure properties and developments in less design cycles. Necessary is a tool that allows the creation and validation of a process flow, taking all the company knowledge into account and delivering a process flow that is ”manufacturable”. This necessities have been introduced and reported in [HWPB03] and [PWOB04].

The analysis of the state-of-the-art silicon based MEMS process development practice revealed the following findings:

- Engineers often consider process development an ”art” without a real structural approach. They require views from different angles to the same results which is not supported by any existing software system.
- Many experiments are done to solve the problems on the way and explore possibilities, dependencies, combinations etc.
- Many results are acquired and are used to solve the next step; results are non-systematically stored in diverse formats (e.g. Excel, Text files, paper, etc.) or merely only in the minds of the process engineers

This causes that results are often not stored in a way to enable reproducibility or retrievability, that no proper reference base is created (if good, kept in spreadsheets sometimes only in the minds of the researchers), that developments are not transparent and that their status is not easily retrievable. Additionally a structured approach for the development sequence and the Design of Experiments (DoE) is missing. Therefore there is the need to establish a suitable structured approach and use a centralized software environment. Such an environment needs to be able to manage process steps and process flows with means to create, maintain, simulate and optimize step and flow recipes; to keep track of the sequence of development steps and support the Design of Experiments; to keep track of all assessment results and make them retrievable. In summary it needs to support Design For Manufacturability, simulation verification and experimental verification.

In summary a structured approach for the process development efforts as well as a system supporting process design and knowledge management is required. These insufficiencies are currently addressed by the PROMENADE [Eur04] project. Its Tracking Environment using a novel concept is introduced in chapter 6.5. The PROMENADE Tracking System is the part to support the experimental verification and knowledge management by addressing reproducibility, control and knowledge conservation and retrieval aspects.

2.4. Summary: Issues addressed by this thesis

Today the silicon based Micro Electro-Mechanical Systems (MEMS) process and design IP industry has inadequate development frameworks. They are not accommodating for all the necessary areas and components required for efficient, reproducible and controlled MEMS IP development. Because MEMS process and design IP development has to cover more disciplines and domains than pure microelectronic design, the resulting frameworks supporting these undertakings need to cover more areas of concern. As such issues have not been addressed, this results in difficulties in process IP transfer, non-systematic storage of product deliverables and development artefacts, incomplete and non-reproducible development steps, insufficient knowledge management and so on.

The product development methodology and the product engineering framework introduced herein proposes a systematic approach to tackle several of these insufficiencies in current research and development (R&D) practices. Many of the issues are especially present in companies using a fabless business model. The thesis provides insight into the identified areas requiring consideration for building up an efficient, reproducible and controlled framework for developing MEMS process and design IP. In summary the following problematic areas are addressed.

2.4.1. Process IP transfer

The issues of process IP transfer are addressed by innovative IP protection and export and import functionalities build into the process development environment. The new concept allows to selectively export and import collected process knowledge to be able to transfer it to customers. More details about this innovative concept is presented in section 6.5.2.

2.4.2. Product Engineering Method

As identified and described in section 2.2.4 and section 2.3.2 there is a need for a specifically tailored product engineering method. This thesis addresses this need and introduces a novel development method for MEMS IP packages. The method is based on two standard methods tailored to the specific needs of MEMS IP development. It is introduced and described in detail in chapter 3.4. To cater for the special needs of the process IP development, a new process development method, described in section 3.5, has been integrated into the overall product engineering method.

2.4.3. Product Engineering Framework

As motivated in section 2.2.5 and section 2.3.3 the new development method motivates the development of a product engineering framework for the MEMS process and design IP industry. This thesis addresses this need by highlighting the requirements for such a framework for this new industry. An exemplary framework addressing all identified requirements will be described in chapter 4. It assembles diverse standard software components into a unique and specifically tailored development framework. The environment also contains several newly developed components featuring the integration of the standard components. The framework is already introduced and used by an industry company.

2.4.4. Process development support

This thesis, together with the other PROMENADE environment components, addresses the need for a structured process development approach. The approach is integrated into the overall product engineering method motivated above.

During the setup of the PROMENADE project the innovative system vision depicted in figure 2.2 has been developed. The figure shows an environment vision for silicon based MEMS process development. Analyzing all sources of process knowledge it can be said that knowledge is generated by engineer discussions and customer interactions, by simulation, by the semiconductor manufacturing and by the intermediate and final silicon assessments. The idea behind the PROMENADE environment is to capture the knowledge from all these sources inside one comprehensive environment, depicted in the middle of the figure. This single box consists of three separate components. A process design component allows the creation and assembly of process steps and flows, performs the Design For Manufacturability (DFM) assessment and coordinates the verification by simulation. This component has been introduced by the thesis of J. Popp [Pop05] and A. Wagener [Wag05]. For the simulation verification the simulation component on the top of the figure is triggered and results are collected back. After successfully performing these two verification steps the process flow is converted into a run card and handed over for manufacturing for experimental verification, shown in the left bottom box of the figure. Additionally the flow is handed over to the Development Tracking component (inside the central box) to coordinate the experimental verification. The Tracking Environment collects all processing and result data from the manufacturing, tests and assessments. The third component inside the central box is responsible for back annotating data and knowledge. It collects data from different sources and stores them inside the tracking or design component. In the future it is foreseen that this component is extended with knowledge generation features analyzing data in the Tracking Environment and generating knowledge for the DFM in the Design Environment.

The complete environment provides for knowledge conservation and knowledge generation. It closes the different loops to enable comparing real results with predicted results and use the differences to recalibrate the knowledge base (see the loop back from the ”Assessment” to the design environment in figure 2.2). Nevertheless the feedback loop from the production lines (red branch in figure 2.2) back into a process design and tracking environment is important as well but mostly covered by existing systems like [Bro05] and [Pho05]. They can be and have to be used to gain this part of the overall necessary data. Existing systems will be interfaced by the Back Annotation component to collect the data from these sources.

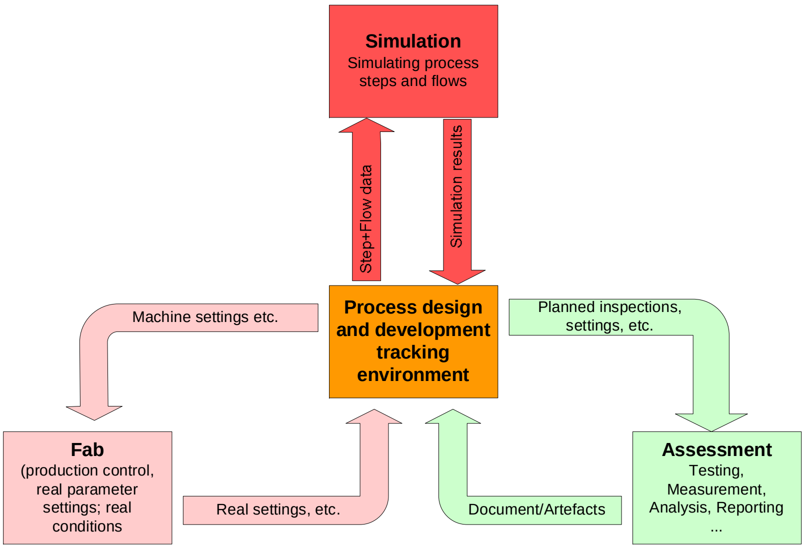

Simulation
Simulating process steps and flows
Step+Flow data
Simulation results
Machine settings etc.
Process design and development tracking environment
Planned inspections, settings, etc.
Fab
(production control, real parameter settings; real conditions
Real settings, etc.
Document/Artefacts
Assessment
Testing, Measurement, Analysis, Reporting
...

Figure 2.2.: Vision of a process development environment

With such a system, a better starting point for new process developments can be provided by making the previously collected knowledge, including rules, process steps and flows, development sequences and all kinds of data files retrievable and make development efforts reportable.

While the Design Environment is covered by the two thesis of J.Popp [Pop05] and A. Wagener [Wag05] the current thesis focusses on the Tracking Environment and the first parts of the Back Annotation component. The novel concepts described in chapter 6.5 enable reproducibility, control, faster time to market and more successful product commercializations.

3. Product Engineering Method

The major challenge to be addressed for setting up Product Engineering activities is to select or define a product development methodology. This is a key activity and depends on the businesses market, its business model and the history of the company, especially whether the company already has adopted a formalized development methodology. As mentioned earlier the context of this thesis is the silicon based MEMS IP industry with a fabless business model. The following descriptions of the Product Engineering method are suited to be used in this context, although they are based on generally applicable methodologies.

To define a suitable Methodology first the needs and the context for new product development efforts need to be summarised. Therefore the following section 3.1 summarises the product development essentials to motivate the features and the approach of the methodology.

The sections after the general perspective describe the newly developed methodology suited to the above summarised business environment. As mentioned earlier the new method consists of a combination of a standard product development method with a project management method. The combination is suited to the needs of the business environment. Before defining the method in detail a brief summary of the evaluated base methods (product development and project management) will be given as well as a reasoning for the selection.

Because one major part of the IP business model, summarised in section 2.2.1 and mainly described in [Fab05], in the MEMS industry is the process/-fabrication IP, the process development methodology is of major importance and will be described in more detail in section 3.5.

3.1. Product development essentials

To define a product development methodology and to set up the necessary method framework, it is essential to recap the essentials for product development. The following sections are only very brief summaries and much more detail can be found in e.g. [Coo98] and [Ulr03].

3.1.1. Drivers for Innovation

First it needs to be pointed out that several different factors drive the innovation and the new product development process. These can be roughly categorized into technology advances, changing customer needs, shortening product life cycles and increased world competition. Additionally a new product can be new to the company developing it or even new to the market. In detail new products can be split into the categories of new to the world products, new product lines, additions to existing product lines, improvements and revisions to existing products, repositionings (products developed for one area but afterwards used in a different area e.g. Aspirin, Viagra) and new products due to cost reductions. A graphical overview about this categorization can be found in [Wik05g]. Because these categories have a major impact on the business case of a potential new product, an idea for a product needs to be carefully analyzed and categorized during the initial assessments.

An additional aspect of innovation is that the best inventor is not necessarily the one who solves a problem best, it is often about seeing and understanding a problem. The detection and understanding of the customers problems is key for an innovative company. Therefore the early involvement of customers in product ideas and a constant contact with the customers to detect problems is key for future innovation. These points are support by the article [Dav04].

3.1.2. Success factors for new product developments

Cooper and his colleagues have analyzed several different product development efforts and approaches. They report, in the widely accepted and supported book ”Product Leadership” [Coo98], the following key ingredients for new product success:

- Embracement of a long-term commitment to product development.
- Development of a vision, objectives and strategy for the new product effort driven by and linked to the business’s corporate objectives and strategy.
- Installation of a systematic, high-quality new product process in the business and to practice discipline, following the principles of the process.
- Making available necessary resources.
- Fostering the innovation in the organization.

These findings can be summarised into the cornerstones of new product performance shown in figure 3.1. It presents that the resource commitment, a new product strategy and a new product process are the major contributors to a businesses new product development performance. The cornerstones Business strategy and resource commitment are beyond the scope of this thesis and highly depend on the business itself. The third cornerstone of a systematic, high quality new product process is addressed by the new development methodology described in section 3.4.

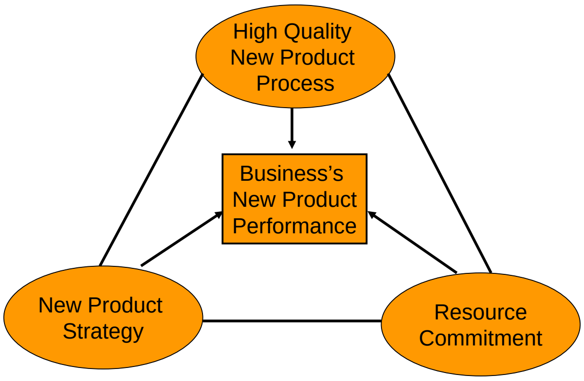

High Quality New Product Process
Business's New Product Performance
New Product Strategy
Resource Commitment

Figure 3.1.: Cornerstones of new product performance

3.1.3. Success factors for a new product process

Before describing the details of the methodology, the key success factors for a new product process or product development methodology should be summarised as they are reported in literature. Coopers book presents several primary and secondary success factors for a new product development methodology. From their point of view a new product process needs to address the following goals:

Exemplary Quality of Execution with focussing on completeness, quality and importance. This can be achieved by establishing quality-control checkpoints, by designating the Leadership team to become quality controllers, by clear and consistent metrics at intermediate checkpoints, by defining activities, tasks, methods and best practices into the development phases, by specifying visible deliverables for each phase, along with action standards and by having effective resource allocation methods for each development phase.

Sharper focus, better project prioritization Often a businesses new product performance suffers from too many projects with not enough resources. Therefore the efforts are not focussed and not properly prioritized. The process needs to allow to have project funnels rather than project tunnels. So there need to be tough (Go/Hold/Recycle/Kill) decisions.

Strong market orientation Marketing tasks need to be designed into the new product process. This includes to perform preliminary market assessments. Market research is to determine user needs and wants, competitive analysis, value-in-use analysis, concept testing, customer reaction and feedback during development, user test and field trials, test market or trial sell and market launch based on a solid marketing plan. Additionally the project executives and leadership team needs to ensure the execution of marketing tasks and provides the necessary marketing resource.

Better up-front homework and sharp, early and stable product definition The up-front homework needs to define the product and its business case in a stable manner. It lies the baseline for a successful or non-successful product development. Therefore the new product process needs to ensure the proper execution of this development phase.

Fast-paced parallel processing (rugby approach) The new product process needs to enable and foster parallel execution of all necessary, multi-functional tasks. Therefore a sequential approach has to be discouraged and each result achieved by one group needs to be passed on to the other groups as soon as possible, although the work might not be finished yet. Exchange of intermediate result and close cooperation is essential and should be fostered by the method.

A true cross-functional team approach The essential ingredients are: A cross-functional team with committed team players; defined team captain or leader, accountable for the entire project; a leader with formal authority (co-opting authority from the functional heads); fluid team structure, with new members added or dropped as work demands; a small core group responsible, committed and accountable from beginning to end.

Products with competitive advantage differentiated products; unique benefits; superior value for the customer; Required ingredients: checkpoint criteria focusing on product superiority; development phases need activities fostering superiority; teams are required to deliver superiority evidence to checkpoints

A fast-paced and flexible process The new product process needs to enable fast developments and needs to accommodate for different types of projects. This implies that smaller or less risky projects need to be able to reduce the overhead by skipping or lightening certain managerial aspects but still execute the same process. The process must allow for flexibility within certain ranges and ensure fluidity in the process execution. For that fuzzy checkpoints need to be used, where decisions to proceed are taken although not all necessary deliverables are completely available but will be within a certain time frame.

The evaluated success factors inducing the above goals and additional factors for the business environment itself supporting the development methodology are given in [Coo98].

3.2. Baseline Product Development Methods

Literature describes a plentitude of different product development methods, often targeted specifically for a certain industry like e.g. the Agile method [Tho05], the V-Model, Extreme Programming (XP) and so on for the software industry. A recent and good summary of different approaches especially for user driven product design is given in the Introduction and Theoretical Framework chapters of [Jan05]. An older article summarizing the different method is [Kar94]. Looking for the common idioms of different approaches, the tendency in most industries is the move to multidisciplinary teams taking the whole life cycle of the product into account. Additionally most modern methods can be categorized into either the family of linear staged methods based on activities or into the group of iterative, information and decision based methods. The linear staged methods have a relatively fixed set of stages executed once as depicted in figure 3.3 with the Stage-Gate™ example. Each stage itself can be of iterative nature. The iterative methods cycle the whole development loop several times as depicted in figure 3.2 with the IPPD example. One example for the linear staged methods is the Stage-Gate™ method introduced in e.g. [Coo98] and other publications of this group of authors. The Integrated Product and Process Development (IPPD) method is an example for an iterative method. The history of both of these method families is summarised in [And96]. Both methods are described in more detail in the following sections.

An interesting background article on product development methods combined with a brief history summary is documented in [LÖ5]. In that publication the author compares Leonardo Da Vincis problem detection and solving approach with current product development approaches and discovered several similarities. The modern method used for the comparison is similar to the Stage-Gate™ method described below and fosters the multidisciplinary team structures. This criteria has been fulfilled by Leonardo Da Vinci via acquiring knowledge from multiple domains. An additional similarity is the multiplicity of available tools at hand to the development crew. Both approaches consider the problem solving or product development as a combination of art and engineering and to foster the creativity this multiplicity of tools is required.

3.2.1. IPPD/CE

The Integrated Product and Process Development (IPPD) approach is an expansion of the concurrent engineering (CE) approach. Both utilize a systematic approach to the integrated, concurrent development of a product and its associated manufacturing and sustainment processes. Their major aim is to satisfy the customer needs.

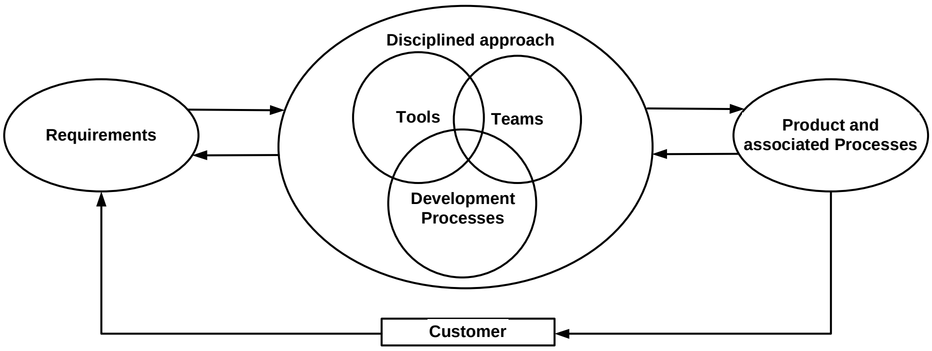


Figure 3.2.: A generic IPPD Iterative Process

IPPD is defined as: ”A management process that integrates all activities from product concept through production and field support, using a multifunctional team, to simultaneously optimize the product and its manufacturing and sustainment processes to meet cost and performance objectives.” [Dep96]

A graphical overview of the generic IPPD management process is presented in figure 3.2. The input to the iterative development process shown in the figure are the requirements generated by the customer(s). They are elicited in a negotiation among many parties, each having serious and important concerns and impacts on the product. For that the customer(s) and their needs need to be known in detail. The disciplined approach includes five general activities: understanding the requirements, outlining the approach, planning the effort, allocating resources, and executing and tracking the plan. Decisions made using this approach should be re-evaluated as a system matures and circumstances (budgetary, threat, technology) change. The disciplined approach provides a framework for utilizing Tools, Teams, and Development Processes in a structured manner. Tools include documents, information systems, methods, and technologies that can be fit into a generic, shared framework that focuses on planning, executing and tracking. Teams are made up of everyone who has a stake in the outcome or product of the team, including the customer and suppliers. Collectively, team members should represent the know-how needed and have the ability to control the resources necessary for getting the job done. Development Processes are those activities which lead to both, the end product and its associated processes. To ensure efficient use of resources, it is necessary to understand what activities are necessary and how they affect the product and each other. Examples include requirements analysis, configuration management, and detailed design drawings. Product and Associated Processes include what is produced and provided to the customer. These steps are iterated if additional or changed customer requirements are encountered.

The key tenets of IPPD are as follows [Dep96]:

- Customer Focus
  The primary objective of IPPD is to satisfy customer needs better, faster and at less cost. The customer needs should determine the nature of the product and its associated processes.

- Concurrent Development of Products and Processes
  Processes and their supported products should be developed concurrently. It is critical that the processes used to manage, develop, manufacture, verify, test, deploy, operate, support, train people, and eventually dispose of the product be considered during development. Product and process design and performance should be kept in balance.

- Early and Continuous Life Cycle Planning
  Planning for a product and its lifecycle processes should begin early in the science & technology phase (especially advanced development) and extend throughout the product's life cycle. Early life cycle planning, which includes customers, functions, and suppliers, lays a solid foundation for the various phases of a product and its processes. Key program events should be defined so that resources can be applied and the impact of resource constraints are better understood and managed.

- Encourage Robust Design and Improved Process Capability
  Encourage use of advanced design and manufacturing techniques that promote achieving quality through design, products with little sensitivity to variations in the manufacturing process (robust design) and focus on development process capability and continuous development process improvement. Utilize such tools as ”Six-Sigma” process control and lean/agile manufacturing concepts to advantage.

- Event Driven Scheduling
  A scheduling framework should be established which relates program events to their associated accomplishments and accomplishment criteria. An event is considered complete only when the accomplishments associated with the event have been completed as measured by the accomplishment criteria. This event-driven scheduling reduces risk by ensuring that product and process maturity are incrementally demonstrated prior to beginning follow-on activities.

- Multidisciplinary Teamwork
  Multidisciplinary teamwork is essential to the integrated and concurrent development of a product and its processes. The right people at the right place at the right time are required to make timely decisions. Team decisions should be based on the combined input of the entire team (e.g. en-

gineering, manufacturing, test, logistics, financial management, contracting personnel) to include customers and suppliers. Each team member needs to understand his/her role and support the role of the other members, as well as understand the constraints under which other team members operate. Communication within teams and between teams should be open with team success emphasized and rewarded.

- Empowerment
  Decisions should be driven to the lowest level commensurate with risk. Resources should be allocated at levels consistent with authority, responsibility, and the ability of the people. The team should be given authority, responsibility, and resources to manage their product and its risk commensurate with the team's capabilities. The team should accept responsibility and be held accountable for the results of their effort. Typically the management and the project leader agree on these deliverables in a kind of ”contract”.

- Seamless Management Tools
  A framework should be established which relates products and processes at all levels to demonstrate dependency and interrelationships. A single management system should be established that relates requirements, planning, resource allocation, execution, and program tracking over the product's life cycle. This integrated approach helps to ensure that teams have all available information thereby enhancing team decision-making at all levels. Capabilities should be provided to share technical and business information throughout the product life cycle. This can be achieved by the use of acquisition and support databases and software tools for accessing, exchanging, and viewing information.

- Proactive Identification and Management of Risk
  Critical cost, schedule and technical parameters related to system characteristics should be identified from risk analysis and user requirements. Technical and business performance measurement plans, with appropriate metrics, should be developed and compared to best-in-class industry benchmarks. This is to provide continuing verification of the degree of anticipated and actual achievement of technical and business parameters.

In summary IPPD and CE are development processes highly focussed on the customer requirements and fulfilling them by iterative product extensions and releases. An overview of the approach and its application in the acquisition process of the Department Of Defence is given in [Dep96]. More resources about the IPPD approach are nicely summarised in [Sof02].

3.2.2. Stage-Gate™ method

The Stage-Gate™ process is an operational roadmap for driving new product projects from idea to launch - a blueprint for managing the new product process improving effectiveness and efficiency. Its central point is a staging and gating process where each stage is executed by a multidisciplinary team. Pioneered and developed by Dr. Robert G. Cooper, published in the book [Coo98] and several papers and websites [Sta05a], [Pro05a], it is the world's most widely used new product development process. Example users are Hewlett-Packard, Procter&Gamble and SonyEricsson. An overview of the generic process is presented in figure 3.3.

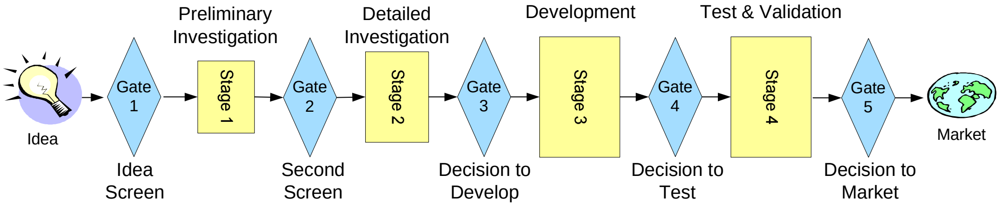


Figure 3.3.: The generic Stage-Gate™ Process

A successful new product development begins with an idea and ends with the launch of a new product. The steps between these points can be viewed as a dynamic process. Stage-Gate™ divides this process into a series of activities (stages) and decision points (gates) covering the success factors described in section 3.1.3.

Each stage contains a set of defined concurrent activities, incorporating industry best practices. Activities during each stage are executed in parallel to enhance speed to market. This is further enhanced by the use of cross-functional/multidisciplinary teams all working towards the same goals and the same set of deliverables required for the next gate. As the method is a blueprint for new product processes, the amount of stages can vary between different process implementations in different environments. Typical process implementations use growing sizes of stages where each later stage is bigger than a previous one in terms of commitment, effort, money etc. On the other hand the risks and the uncertainties of the project need to decrease.

Gates are the decision points where the Gatekeepers (described below) decide whether to continue funding a project. The gates consist of three main elements which are the required deliverables, a set of evaluation criteria and the outputs. The outputs, in the positive assessment case, consist of the funding decision, an agreed action plan for the next stage and an agreement on the set of required deliverables for the next gate. At the gates selection criteria are applied and tough decisions are made about the projects but the gates are essential for the success of a new project strategy. They are the key to distinguish between the potentially good and bad products and enable to focus scarce resources on the potentially best projects, thus establishing value for money. The outcome of the decision point is either to Go(continue developments), Kill(cancel developments), Hold(temporarily stop developments), or Recycle(redo previous stage) the project. Gates also act as ”quality control” checkpoints evaluating among other things: Have the deliverables been generated in a quality fashion? Is the project (still) attractive from an economic and business standpoint? Is the information sufficient to make a decision? Is the action plan and request for resources sound?

The gates are executed by the Gatekeepers of the process. The gatekeepers are a team of senior leaders who make Go/Kill/Hold/Recycle decisions at the gates. They are from different functional areas and can commit the required resources. They use a pre-set list of criteria and rules for the gates. By those rules and criteria they are bound so that they can not play favourites.

The benefits of using Stage-Gate™ are that it introduces discipline into an ordinarily chaotic process. It focuses attention upon quality of execution and speeds up the new product development process. The method ensures a complete process (no critical steps are omitted) and facilitates a focus on the performance of a new product. All this results in a faster, more effective and efficient process that produces winning new products. Software environments supporting and guiding developments based on this method are available as well, one example can be found on [DRM05b].

3.2.3. Conclusions

As regularly reported in literature, there is nothing like ”The method” fitting all purposes and industries. Therefore the application area of the product development method needs to be carefully taken into account. The method developed in the scope of this thesis is to be used in the silicon based MEMS process and design IP area. Advantages of the one or the other method may be less relevant in the considered case. As an example, the criticism about the staged development given in [And96] about the missing integration of all life cycle aspects (development, manufacture, distribution, support and the early inclusion of suppliers) is less relevant to the IP industry. This is due to the fact that e.g. the manufacturing is only of indirect relevance, the area of distribution is not required and the inclusion of suppliers into the development is less of an issue because of its limited applicability. These points - being the major differentiators of IPPD - can sufficiently be covered by a staged development as well.

Another specialty to take into account is the sequential nature of the MEMS process developments. Within these efforts the steps material development, material optimization, sequencing, modularisation, etc. need to be executed in this standard sequence and can hardly be changed or parallelized. Therefore the process IP development has a kind of predefined nature and needs to be incorporated. These efforts can be integrated more easily into the Stage-Gate™ method.

The MEMS process IP development has a very high research character. Due to that the ”contract” between the management and the development team, being the essential base in IPPD, is very difficult to achieve because of all the unknowns. Additionally, considering a fabless startup scenario, the methodology needs to cater for structure, possible management involvement, team building and multi-site development. Therefore a highly integrated development method like IPPD requiring a co-located development team and offering only less management involvement possibilities has several disadvantages. Especially the required multi-site development team of the fabless approach, the process developments needs to be done at least at one place with equipment access, requires a more structured and controlled approach.

Therefore it can be concluded that the advantages of IPPD over Stage-Gate™ reported in [And96] are less applicable and have some drawbacks as well, especially to the MEMS IP development. As the author suggests in the showcase, the transition from a Stage-Gate™ method to an IPPD based method is possible as well and to structure the initial chaos, a staged method is preferable. Additionally Stage-Gate™ has a very good reputation, is in widespread use and is supported by various additional authors [Ols04], [Bru05], [Dav04] and [VTT01] pages 59-71. Because of that the decision was taken to base the herein proposed methodology on the Stage-Gate™ approach, having the preconditions for IPPD in mind.

3.3. Baseline Project Management Methods

The necessity for the addition of a Project Management (PM) method extending the Product Development method is described in the methods overview (section 3.4.2) and the Introduction. To not digress the subject to much, no general introduction into project management itself is given herein. For such introductions and the historical development of project management [Wik05j] and [PMI05] can be reviewed. Especially [PMI05] gives a good definition for what a project is, which success factors do exist and what distinguishes a project from a product, programm and operations.

A literature study revealed that there exists a variety of different PM Methods. An exhaustive list of such methods and project maturity models can be found in [PMF05]. An analysis of how commonly the different methods are used and their general applicability to product developments of all kinds, not limiting it to software projects, reduces the possible candidates. Additionally it must be easily possible to combine the PM method with a product development method. Applying all these criteria the remaining list consists of PMBOK [Ins04], PRINCE2 [GC02] and ISO10006 [Int05b].

To give an overview, the following sections very briefly summarise these three methods and the differences between them. Advantages and disadvantages will be discussed in the Conclusion section 3.3.4.

3.3.1. PMBOK

The Project Management Body Of Knowledge (PMBOK) is standardized in the IEEE Std. 1490-1998 and can be considered *the* project management standard in North America. It is officially documented in [Ins04] and a good introduction can be found in [Pro05b].

Under PMBOK a project is initiated, guided, etc. by its Stakeholders'. These are the people involved or affected by the project activities e.g. project sponsor, project manager, user, suppliers, customers. They define the needs, provide resources and so on. The PMBOK project management principles use five main PM processes which are initiating, planning, executing, controlling and closing of the project. The process partly overlap each other as well as they interact with each other. On overview of the interaction of the processes is presented in figure 3.4. PMBOK describes the processes in terms of Inputs (e.g. documents, plans, designs), tools & techniques (mechanisms applied to the inputs) and outputs (e.g. documentations, products).

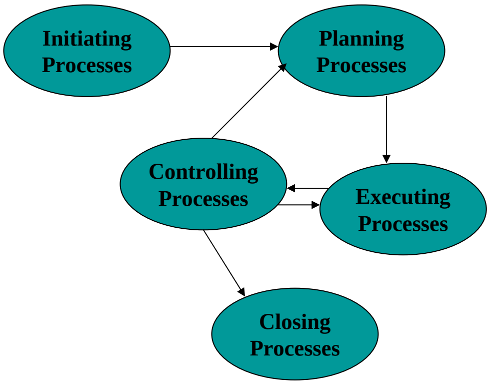


Figure 3.4.: Management process of the PMBOK

Additionally the standard defines nine knowledge areas which are closely tied into the process and the tools, techniques and best practices. The nine knowledge areas are heading the different rows in table 3.1 and the columns indicate the different process groups. The cross sections of the table list the corresponding processes of the group in that knowledge area and refer to the corresponding PMBOK chapter. A graphical overview of the interaction of the different knowledge areas is presented in figure 3.5.

In summary PMBOK can be characterized to be a collection of processes and knowledge areas generally accepted as best practices within PM. Its basic concepts are applicable to all projects and programs (and operations). The knowledge base provides the fundamentals of PM, irrespective of the type of project (software, construction, environmental, aerospace, etc.) and recognizes five basic process groups and nine knowledge areas typical of almost all projects.

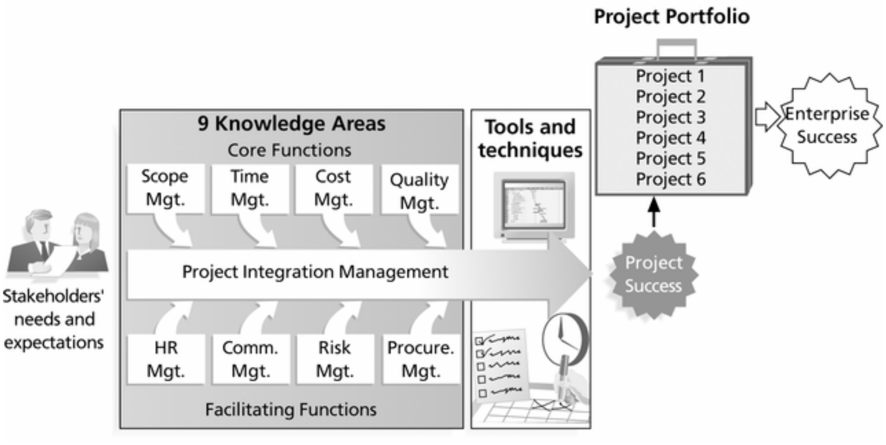


Figure 3.5.: Knowledge areas and their interaction of the PMBOK (extracted from [PMI05])

Knowledge Areas & Chapter numbers	Process Groups & Processes Within Groups Process section no. within PMBOK Chapter				
	Initiation	**Planning**	**Executing**	**Monitoring & Controlling**	**Closing**
Project Management Integration 4	1 Develop Project Charter 2 Develop Preliminary Project Scope Statement	3 Develop Project Management Plan	4 Direct and Manage Project Execution	5 Monitor and Control Project Work 6 Integrated Change Control	7 Close Project
Project Scope Management 5		1 Scope Planning 2 Scope Definition 3 Create WBS		4 Scope Verification 5 Scope Control	

Knowledge Areas & Chapter numbers	Process Groups & Processes Within Groups Process section no. within PMBOK Chapter				
	Initiation	**Planning**	**Executing**	**Monitoring & Controlling**	**Closing**
Project Time Management 6		1 Activity Definition 2 Activity Sequencing 3 Activity Resource Estimating 4 Activity Duration Estimating 5 Schedule Development		6 Schedule Control	
Project Cost Management 7		1 Cost Estimating 2 Cost Budgeting		3 Cost Control	
Project Quality Management 8		1 Quality Planning	2 Perform Quality Assurance	3 Perform Quality Control	
Project Human Resource Management 9		1 Human Resource Planning	2 Acquire Project Team 3 Develop Project Team	4 Manage Project Team	
Project Communications Management 10		1 Communications Planning	2 Information Distribution	3 Performance Reporting 4 Manage Stakeholders	
Project Risk Management 11		1 Risk Management Planning 2 Risk Identification 3 Qualitative Risk Analysis 4 Quantitative Risk Analysis 5 Risk Response Planning		6 Risk Monitoring and Control	
Project Procurement Management 12		1 Plan Purchases and Acquisitions 2 Plan Contracting	3 Request Seller Responses 4 Select Sellers	5 Contract Administration	6 Contract Closure

Table 3.1.: PMBOK overview

3.3.2. PRINCE2

The PRINCE2 project management method, published and maintained by the Office of Government Commerce (OGC) in the UK, is used in many different companies and has become the standard for project management in Europe. The OGC has published several books on PRINCE2 where [GC02] is the most important one. Additional good introductions are [Spo05] and [Wik05h] and a handbook putting it into practice is [Ben02]. The method is based on the

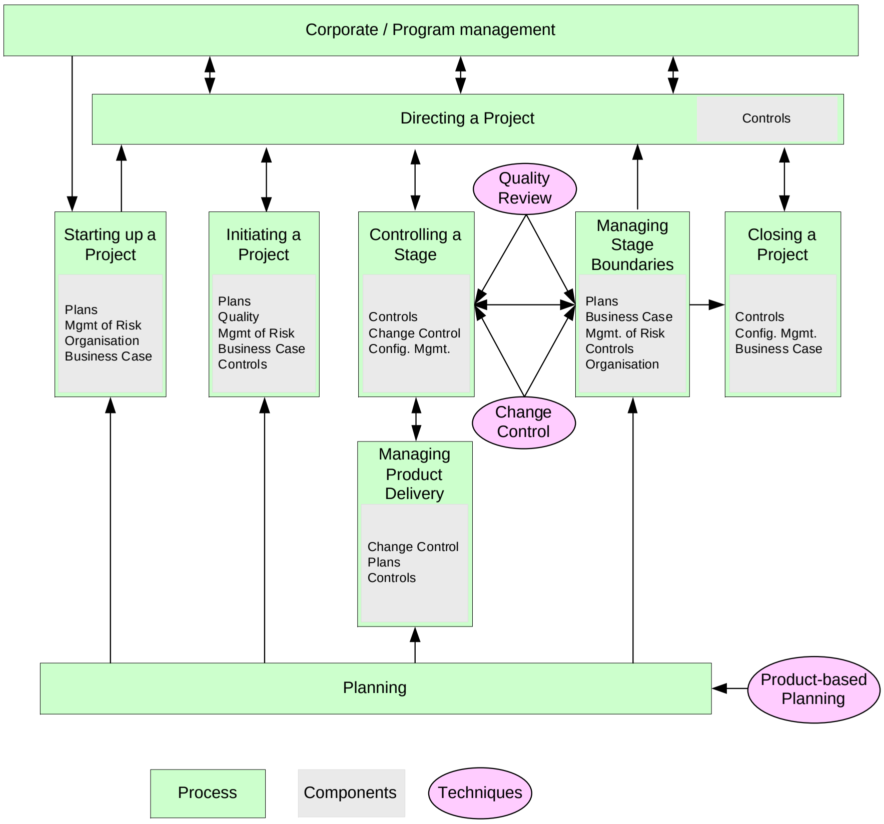


Figure 3.6.: Management processes, components and techniques of PRINCE2

approach to treat the efforts of project management as a process. It is highly product oriented by focussing on the business case and emphasizes the quality aspects.

The PRINCE2 method provides a framework of components and techniques to perform the different project management processes. It is a general PM method not limited to the management of IT projects, although it originates from this area. The controlled and systematic preparations and execution of projects in a process following PRINCE2 explicitly prevents problems of other PM methods with weaker processes. This is a central aspect and an additional benefit compared to other methods. An overview about the PRINCE2 process, its components and the techniques provided is presented in figure 3.6. The figure shows the interaction between the different processes and how they tie into each other. Additionally the grey boxes inside the process squares show the components used in that process and its elements of the process. The ellipses show the techniques coming with the method and in which process they are envisioned to be used.

The PRINCE2 benefits can be summarised as (1)identifies management, specialist and quality Products / Deliverables and helps to ensure that they are produced on time and to budget (2)focuses attention on the quality of Products/Deliverables and on the viability of the Business Case (3)separates the management and specialist aspects of Organisation, Planning and Control (4)facilitates controls at all levels (5)makes the project's progress more visible to management (6)provides a communication medium for all project staff (7)ensures that work progresses in the correct sequence (8)involves senior management in the project at the right time and in the right place (9)allows the project to be stopped and, if required, re-started completely under management control, at any time in the project's life (10)is in the Public Domain and requires no license fee (11)has a well established User Group dedicated to the support, promotion and strengthening of the method

In summary PRINCE2 can be characterized to be a collection of processes, components and techniques generally applicable to PM in diverse fields (software, construction, environmental, aerospace, etc.). Its basic concepts are applicable to all projects and programs (and operations). The method provides the fundamentals of PM, recognizing eight basic processes with subelements, eight components used during the different processes and three basic techniques applied.

Process	**Process elements**	**Components**	**Techniques**
Starting up a Project	SU1. Appointing a Project Executive and a Project Manager SU2. Designing a Project Management Team SU3. Appointing a Project Management Team SU4. Preparing a Project Brief SU5. Defining Project Approach SU6. Planning Initiation Stage	Plans Mgmt. of Risk Organisation Business Case	
Directing a Project	DP1 Authorising Initiation DP2 Authorising a Project DP3 Authorising a Stage or Exception Plan DP4 Giving Ad Hoc Direction DP5 Confirming Project Closure		
Planning	PL1. Designing a Plan PL2. Defining and Analysing Products PL3. Identifying Activities and Dependencies PL4. Estimating PL5. Scheduling PL6. Analysing Risks PL7. Completing a Plan	Product-based Planning	
Initiating a Project	IP1. Planning Quality IP2. Planning a Project IP3. Refining the Business Case and Risks IP4. Setting Up Project Controls IP5. Setting Up Project Files IP6. Assembling a Project Initiation Document (PID)	Plans Quality Mgmt. of Risk Business Case Controls	

Process	Process elements	Components	Techniques
Controlling a Stage	CS1. Authorising Work Package CS2. Assessing Progress CS3. Capturing Project Issues CS4. Examining Project Issues CS5. Reviewing Stage Status CS6. Reporting Highlights CS7. Taking Corrective Action CS8. Escalating Project Issues CS9. Receiving Completed Work Package	Controls Change Control Config. Mgmt.	Quality Review Change Control
Managing Product Delivery	MP1. Accepting a Work Package MP2. Executing a Work Package MP3. Delivering a Work Package	Change Control Plans Controls	Quality Review Change Control
Managing Stage Boundaries	SB1. Planning a Stage SB2. Updating a Project Plan SB3. Updating a Project Business Case SB4. Updating the Risk Log SB5. Reporting Stage End SB6. Producing an Exception Plan	Plans Business Case Mgmt. of Risk Controls Organisation	
Closing a Project	CP1. Decommissioning a Project CP2. Identifying Follow-on Actions CP3. Project Evaluation Review	Controls Config. Mgmt. Business Case	

Table 3.2.: PRINCE2 overview

3.3.3. ISO10006

Another standard in the context of project management is ISO10006 which is document in [Int05b]. Up to now the standard has been released in two revisions, ISO10006:1997 and ISO10006:2003 [Bri03]. The initial version of this standard in based on a draft of PMBOK and is therefore relatively similar to PMBOK. But the ISO standard addresses project management from the quality aspects and targets more to quality assurance in project management. On top of that it is less detailed than PMBOK and offers less help to project managers. Being based on PMBOK it shares the weakness of being more a collection of best practices, knowledge areas and required skills. The ISO10006 focusses less on an explicit project management process and has only a very abstract and loose project execution process. Therefore it is less practical and provides more a kind of a checklist for quality driven project management than a project management process. An additional weak point of the ISO10006 is that it misses out or only very briefly covers subjects like scope, cost, time and risk. Literature comparing PMBOK and ISO10006 ([THE95]) or analyzing ISO10006 itself ([Pro05c]) discourage the use of ISO10006. The later article even states that the application of ISO10006 could result in the opposite effects because of potential misinterpretations of some guidelines having no clear context. Because of these weaknesses and the very limited usage of ISO10006 in commercial environments this method is disregarded from further investigation.

3.3.4. Conclusions

Only PMBOK and PRINCE2 remain for consideration as a baseline project management method, after having eliminated the ISO10006 standard because of its weaknesses and its focus on quality management in projects. Investigating these two possibilities further it can be noted that PMBOK is less process centric, although it defines the five basic processes. PMBOK puts a lot of emphasis on the different knowledge areas a project manager needs to handle and approaches project management from the project managers required skill set. This approach makes the PMBOK method very valuable for teaching project management skills. In contrast PRINCE2 is focussed more on the practical guidance of a particular project. This point of view is shared by [Wid05] and [Ver05]. This process and guidance orientation, together with the inherent management of the project in stages, makes PRINCE2 a better fit with the Stage-Gate™ method than PMBOK.

An additional difference between PMBOK and PRINCE2 is, that PRINCE2 emphasizes the business case and the project brief of a project and PMBOK does not even mention one of them. The business case is a very central vehicle for the PRINCE2 method. The method enforces the constant update according to gained knowledge and changing conditions as well as its constant verification by the project board after each development stage. Therefore PRINCE2 has a higher applicability for product developments than PMBOK. This is even emphasized further by the product, rather than task, centric planning method of PRINCE2. A product in that context can be any kind of outcome of a project, a tangible one, like a paper document, or an intangible one like a redefinition of a business process. On the other hand PMBOK contains knowledge areas like HR and Procurement Management which are not covered in PRINCE2.

Another distinguishing factor is that PRINCE2 is designed for a variety of customer and supplier scenarios while PMBOK primarily considers that the project is run by one supplier for one customer. This factor makes the PRINCE2 method more flexibly applicable, especially for scenarios where the project is run by an instance in a sandwich position having customers and external suppliers.

Furthermore it can be noticed that PRINCE2 puts more emphasis on documentation than PMBOK. It provides a plentitude of templates or guidelines for project templates helping to become the knowledge pool for a project. Between those are the definitions of an Issue Log, a Risk Log and Acceptance Criteria which are not addressed by PMBOK. On the other hand this documentation emphasis can burden a project and potentially makes the management overhead for small project significant. This can be prevented by using PRINCE2 as a framework where a project can select the applicable standards depending on risk and size.

Summing up and concluding, it can be said that PRINCE2 is a better extending match for the Stage-Gate™ method than PMBOK. This is due to its advantages in its process model, its product and business case centric definition and its more flexible project environment scenario. Its drawbacks of missing out certain areas, which are covered by PMBOK, can be circumvented by adding not covered ground from the PMBOK and using its more detailed documentation of best practices. This combined approach is suggested by [Ver05] and [Sie05b] as well. The heaviness on documentation can be limited by relaxing the constraints for small and minor risk projects. On the other hand this documentation focus builds a good baseline for quality certification, as described in chapter 5.

3.4. New Product Development Methodology

As described in the Introduction (chapter 1) the newly defined product development methodology is based on a process based approach. It addresses the success factors and goals summarised in section 3.1.2 and is thought to take a product idea through the different development steps until market launch. Term definitions, additional to the ones given in the Glossary, are given in section 3.4.1. The following section 3.4.2 will give an overview about the novel methodology. The method overview will be extended in section 3.4.3 with more details about how the base methods have been integrated and extended and how the developed process guides the MEMS process and design IP development. Additionally the different deliverable of the development stages and the gate criteria will be discussed.

3.4.1. Definitions

The current section provides short definitions of the different terms and items of the method in addition to the terms and definitions given in the Glossary. Additionally it summarises commonalities between the different steps (stages and gates) of the development method to shorten the descriptions of the individual stages and gates.

Deliverable The term deliverable refers to any kind of produced result, typically stored in a file, produced during a development stage and required for the next development gate. Deliverables can be documents of diverse type (MS Word, Adobe PDF, etc.), spreadsheets, presentations, test results, databases, software etc. A deliverable might become a part of a product or might be a stepping stone to progress the product development project. In the current context the term Deliverable is used as a synonym for the PRINCE2 term product.

Stages Each stage of a development project contains a set of defined concurrent activities, incorporating industry best practices. Activities during each stage are executed in parallel to enhance speed to market and are focused on the deliverables of the development stage. To foster the focussing onto the deliverables a deliverable oriented, not activity oriented, planning process is used to establish the stage plans. The speed to market is further enhanced by the use of multidisciplinary teams all working towards the same goals and on the same set of deliverables required for the next gate. The stages are primarily focussed to produce high quality deliverables to pass the next gate only. Typical stages have growing sizes during the lifetime of a project where each later stage is bigger than a previous one in terms of commitment, effort, money etc. On the other hand the risks and the uncertainties of the project have to decrease to enable successful assessment during the gates.

Gates As introduced in section 3.2.2 gates are the decision points in the lifetime of a project where the Gatekeepers decide whether to start or continue funding a project. The gates are the key to distinguish between the potentially good and bad projects or products. At the gate the gatekeepers take tough decisions about the projects. The decisions are essential for the success of a new product strategy. Essentially they enable to focus scarce resources on the potentially best development projects, thus establishing value for money. The outcome of these funnel decisions is either to Go, Kill, Hold, or Recycle a project. Gates consist of three main elements which are the required deliverables, a set of evaluation criteria and the outputs. The required deliverables document the advancement of the project. They are based on a predefined catalogue per stage given by the development method and are agreed between the gatekeepers and the project management in the preceding gate. The evaluation criteria are used to assess the project. They typically consist of knockout questions like strategic fit used to weed out not fitting products and a set of should-meet criteria scoring the project. The overall scoring of a project is used for prioritizing the different projects. The outputs of a gate, in the positive case, consist of the funding decision, an approved action plan for the next stage and a list of deliverables and criteria for the next gate.

The gate assessments are divided into two phases, the checking of a couple of killer variables or must-meet criteria and the checking of should-meet criteria by a scoring model check. The first phase of the project assessment filters out the not viable, too low quality or strategically not fitting projects. Only projects passing all criteria defined for the gate with ″Yes″ or ″Not Applicable″ will be assessed further with a scoring check. The failing projects will be put on ″Hold″, on ″Recycle″ to improve quality or will be killed altogether depending on the estimated potential. This assessment is regularly applied to all projects which are on ”Hold” to move a project to development, keep on ”Hold” or to kill a project.

The second phase of the gate performs a scoring model check of the project and determines the projects prioritisation. For that each gatekeeper scores the project with the predefined should-meet criteria checklist for that gate and assigns a ”point” value to each of the criteria. The results of all gatekeepers are averaged and if the value for one criteria is below threshold or the overall threshold is not met, the project will be put on ”Hold” or ”Recycle”. Projects which are on ”Hold” will be reassessed later again while projects set to ”Recycle” need to execute the previous stage again to achieve sufficient quality of the deliverables. Depending on the overall scoring of the project, it might be even necessary to compare the assessed project with other projects under development. In cases of limited resources and outstanding scoring of a project, another project might be put on hold at its next gate and the resources might be reallocated to the development of the outstanding project. Depending on the progression of the prioritized project the situation can be reversed again later. These mechanisms allow to relatively dynamically allocate resources between projects to maximize the potential value for money. But it should be noted that the resource allocation for one stage remains constant or increases. Decreasing of resource during a stage is not foreseen because that might surrender results of the shrunken project useless and might make the restart of the project impossible.

Gatekeepers The gates are executed by the Gatekeepers of the process. The gatekeepers are a team of medium or senior leaders taking Go/Hold/Recycle/Kill decisions at the gates. They are from different functional areas and have to be entitled to commit resources to a project. They use a pre-set list of criteria and rules for the gates to prevent promoting ”personal pet projects”. By those rules and criteria they are bound so that they can not play favourites. The first three gates of the process can be performed by lower level management, one design, one process and one modelling group leader, the CTO and a marketing representative. Later gates will be performed by the VPs of the different business areas, including financial and other involved business groups.

3.4.2. Method overview

Several product development and project management methods reported in literature have been examined as described in more detail in section 3.2 and 3.3. The analysis revealed that the combination consisting of the product development method called Stage-Gate™, introduced in [Coo98], and the PRINCE2 project management method (documented in [GC02]) provides a good starting point for defining a development method for the intended environment. Both methods are based on the approach to treat the efforts (product development and project management) as a process, emphasize the quality aspects and are used by a widespread commercial and governmental community. A sketch of the combined processes is presented in figure 3.7.

The combination of the Stage-Gate™ product development methodology and the PRINCE2 project management method is especially useful. This is because Stage-Gate™ focuses on the efficient guidance from the product idea to a marketable product. Therefore is defines the required deliverables and the evaluation criteria to assess the viability of the product idea after the different R&D stages. By that it focuses mainly on the ”What?” and the principle steps of the ”How?” of successful product design. The development is based on several phases or stages (marked yellow in figure 3.7). Potentially iterations are executed inside the stages while each stage ends in a gate (marked blue in figure 3.7). Apart from quality planning and other quality improvement techniques, the quality aspects are built into the gates between the development stages. During the gate assessments the deliverables produced in the previous stage are evaluated according to pre-defined criteria, agreed upon in the previous gate. Depending on this assessment, management takes the Go/Hold/Recycle/Kill decisions. Additional quality efforts are performed in parallel to the complete project (marked blue in figure 3.7). Their execution is ensured by the project management with the support of the quality assurance department.

The PRINCE2 method is more intended to support efficient and focussed project management in a process and quality fashion. It provides a framework of tools and deliverables to perform the different project management processes. Therefore the method mostly deals with the details of the ”How?” and builds a good addition to the Stage-Gate™ method. The quality aspect is build into the method inherently by providing guidance for audits/gates, quality planning, change control and so forth. The means are built into the different processes OF the development project.

The example process outlined in figure 3.7 consists out of five stages where each stage is bigger than the previous stage in terms of involved effort, money, time, etc. After the initial assessment of a product idea in Gate 1 the first small stage analyses the market potential and the general technical feasibility. The assessment in Gate 2 might take the decision to start-up a project. Stage 2 starts and initializes a development project by appointing the project responsible persons, setting up the project infrastructure, developing the project brief out of the project mandate, setting up an initial business case and developing the different plans according to PRINCE2. This stage is followed by the quick Gate 3, only checking whether the boundary constrains determined during Gate 2 are still valid. If this is the case it leads to the Stage 3 evaluating the product idea in a more detailed technical, market, legal and financial investigation. This includes building a more solid business case and a technical concept prototype

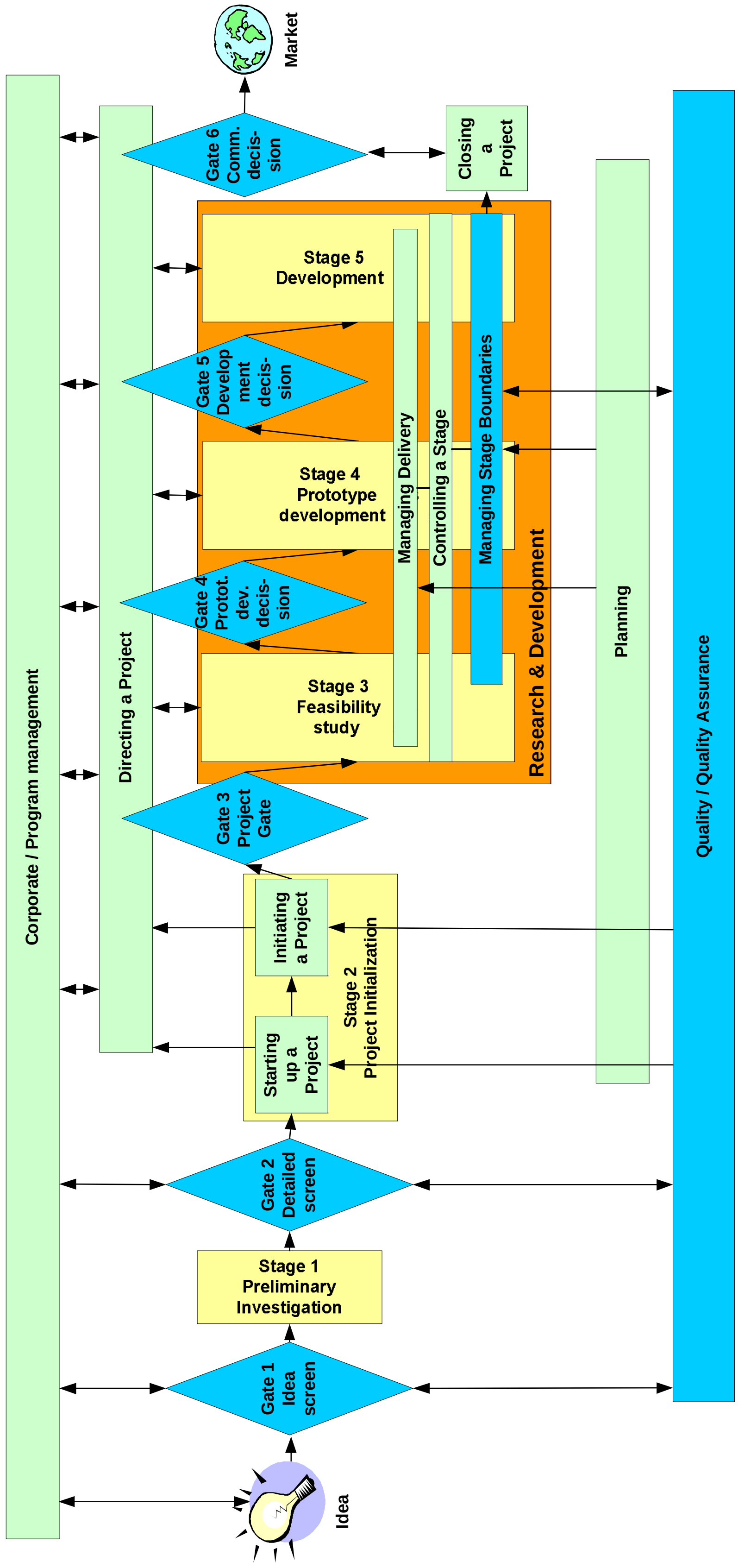

Corporate / Program management
Directing a Project
Idea
Gate 1 Idea screen
Stage 1 Preliminary Investigation
Gate 2 Detailed screen
Starting up a Project
Initiating a Project
Stage 2 Project Initialization
Gate 3 Project Gate
Stage 3 Feasibility study
Gate 4 Protot. dev. decis- sion
Stage 4 Prototype development
Gate 5 Develop ment decis- sion
Stage 5 Development
Managing Delivery
Controlling a Stage
Managing Stage Boundaries
Research & Development
Gate 6 Comm. decis- sion
Closing a Project
Market
Planning
Quality / Quality Assurance

Figure 3.7.: Product Development Methodology overview ([Ort05a])

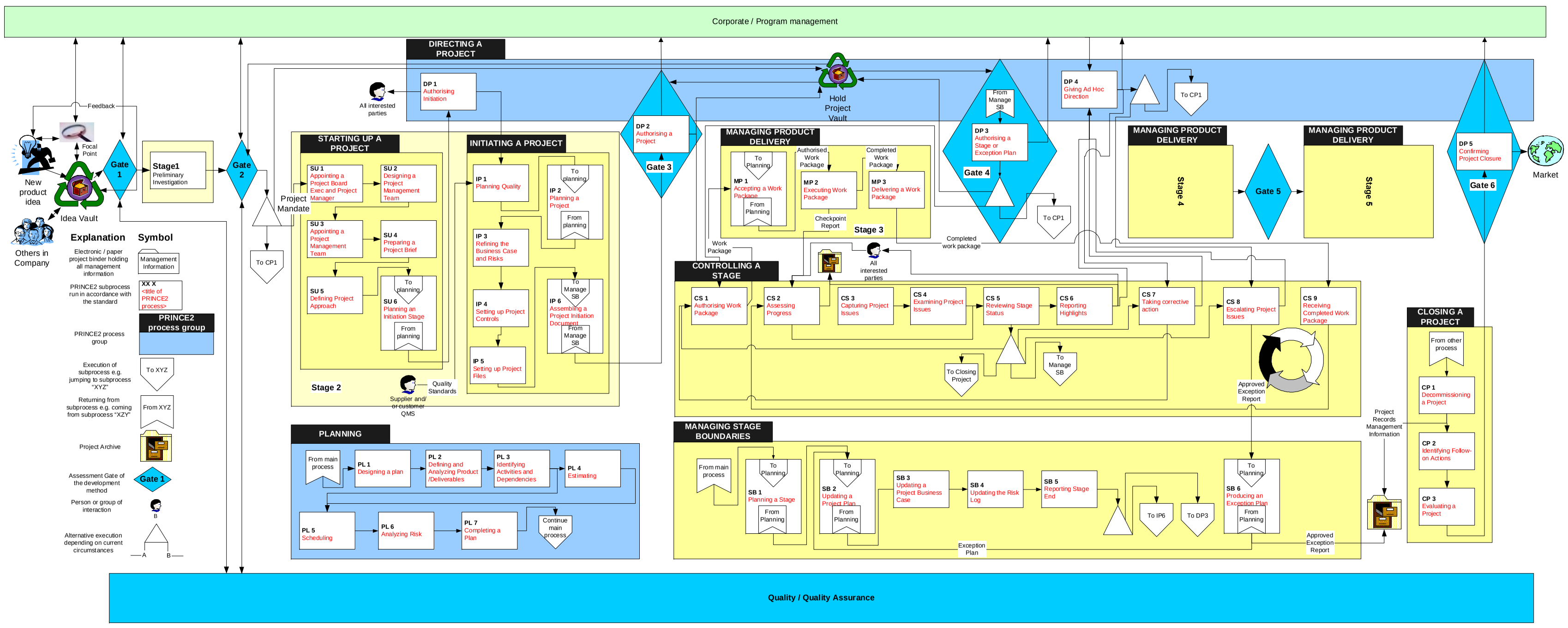
(approx. 10% development effort). Gate 4 builds a critical gate (money gate) before heavily investing into the project in the following stage. This Stage 4 performs the prototype development and contains initial lab testing and delivers a reproducible prototype. Several intermediate quality assessment points e.g. by a quality audit are included in this stage as well. Gate 5 reviews the completeness and the quality of the prototype development. The final development stage continues with intensive Testing, broadening the process windows and wrapping up the other product components in Stage 5. Final Gate 6 takes the decision to market and roll-out the product. The whole progression of these efforts is managed by the PRINCE2 management processes and all gates are potential redo/hold/stop points. This five stage process in figure 3.7 should be considered a minimal example for the stages. Fewer stages would add a significant risk because each stage would become bigger resulting in more commitment to that stage. Company dependent implementations of the processes might foresee more stages than this example or introduce intermediate gates within a stage to evaluate the current status.

3.4.3. Method composition

Section 3.4.2 gave a brief overview about the composed method. The following sections will describe the method in more detail and will adapt the generic base components to the business environment. They will describe which processes and activities are carried out in the different stages, which deliverables are foreseen in which stage and the criteria applied at the gates.

In the merged approach the overall development is guided by the Stage-Gate™ process and all development stages and gates are executed by the PRINCE2 (sub-)processes following the Stage-Gate™ practices. The tasks and main features and characteristics of the stages and gates have been introduced in section 3.2.2. By definition projects managed with PRINCE2 use stages and product based planning focussing on the deliverables of a stage. The method defines mechanisms how to handle deviations and escalation situations and offers several tools and techniques to guide the ventures. Therefore it fits very nicely to the staged overall development approach of the Stage-Gate™ method.

All processes, controls and techniques of the PRINCE2 standard have been integrated into the new method and the management processes are executed according to the standard. Because of that the following descriptions only focus on the steps, the integration of the methods and the development tasks and criteria themselves. The details of the PRINCE2 management processes are only given as far as there are deviations from the standard. For details about the management efforts the literature [GC02] and [Ben02] can be consulted and an overview in the form of a process map is given in figure 3.9. Parts of this figure will be extracted later on for better readability.



Corporate / Program management
Feedback
Focal Point
New product idea
Idea Vault
Others in Company
Gate 1
Stage1
Preliminary Investigation
Gate 2
Project Mandate
To CP1
DIRECTING A PROJECT
DP 1 Authorising Initiation
All interested parties
STARTING UP A PROJECT
SU 1 Appointing a Project Board Exec and Project Manager
SU 2 Designing a Project Management Team
SU 3 Appointing a Project Management Team
SU 4 Preparing a Project Brief
SU 5 Defining Project Approach
SU 6 Planning an Initiation Stage
To planning
From planning
Stage 2
Quality Standards
Supplier and/ or customer QMS
INITIATING A PROJECT
IP 1 Planning Quality
IP 2 Planning a Project
IP 3 Refining the Business Case and Risks
IP 4 Setting up Project Controls
IP 5 Setting up Project Files
IP 6 Assembling a Project Initiation Document
To Manage SB
From Manage SB
DP 2 Authorising a Project
Gate 3
Hold Project Vault
MANAGING PRODUCT DELIVERY
MP 1 Accepting a Work Package
MP 2 Executing Work Package
MP 3 Delivering a Work Package
To Planning
From Planning
Authorised Work Package
Completed Work Package
Checkpoint Report
Stage 3
Work Package
Completed work package
DP 3 Authorising a Stage or Exception Plan
Gate 4
To CP1
DP 4 Giving Ad Hoc Direction
To CP1
Stage 4
Gate 5
Stage 5
DP 5 Confirming Project Closure
Gate 6
Market
CONTROLLING A STAGE
CS 1 Authorising Work Package
CS 2 Assessing Progress
CS 3 Capturing Project Issues
CS 4 Examining Project Issues
CS 5 Reviewing Stage Status
CS 6 Reporting Highlights
CS 7 Taking corrective action
CS 8 Escalating Project Issues
CS 9 Receiving Completed Work Package
To Closing Project
To Manage SB
Approved Exception Report
CLOSING A PROJECT
From other process
CP 1 Decommissioning a Project
CP 2 Identifying Follow-on Actions
CP 3 Evaluating a Project
Project Records Management Information
PLANNING
From main process
PL 1 Designing a plan
PL 2 Defining and Analyzing Product /Deliverables
PL 3 Identifying Activities and Dependencies
PL 4 Estimating
PL 5 Scheduling
PL 6 Analyzing Risk
PL 7 Completing a Plan
Continue main process
MANAGING STAGE BOUNDARIES
SB 1 Planning a Stage
SB 2 Updating a Project Plan
SB 3 Updating a Project Business Case
SB 4 Updating the Risk Log
SB 5 Reporting Stage End
SB 6 Producing an Exception Plan
To IP6
To DP3
Exception Plan
Approved Exception Report
Quality / Quality Assurance
Explanation
Symbol
Electronic / paper project binder holding all management information
Management Information
PRINCE2 subprocess run in accordance with the standard
XX X <title of PRINCE2 process>
PRINCE2 process group
Execution of subprocess e.g. jumping to subprocess "XYZ"
To XYZ
Returning from subprocess e.g. coming from subprocess "XZY"
From XYZ
Project Archive
Assessment Gate of the development method
Gate 1
Person or group of interaction
Alternative execution depending on current circumstances
A
B

Figure 3.8.: Complete development method

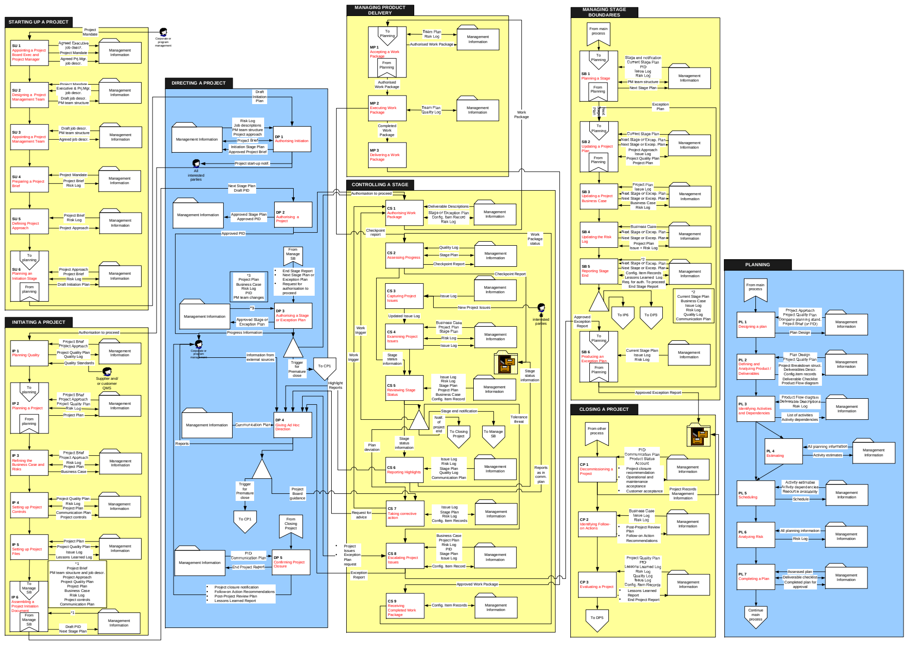


Figure 3.9.: Complete PRINCE2 process map

The fully integrated method with a medium level of information detail is presented in figure 3.8 and a version with a serialized method is presented in figure 3.10. The drawings are described in the following sections in more detail and are cut into two pieces for better readability in figure 3.11, 3.13 and 3.14. Figure 3.11 shows the first phase of the method from the product idea until the assessment after the initial investigation. In figure 3.13 the development project preparations and pre-evaluations are completed. Figure 3.14 shows the R&D stages and gates from the Feasibility stage to the marketable product.

The serialization figure 3.10 abstracts from the belonging of the PRINCE2 process steps to their different process groups. The figure highlights the flow in the optimum case, no major deviations, with green arrows to show the standard flow. The sequences in the exception cases are presented by black arrows. Additionally the graphic shows the process steps grouped into the major phase of preparing a development project, the loop inside a R&D stage, the group of steps finalizing a stage, the gates, the exception handling steps and the project finalization steps with underlaying coloured boxes. The projects research and development efforts are driven by the iteration of the process blocks

- inside a stage loop,
- finalizing a stage and
- assessing the results in the gating phase.

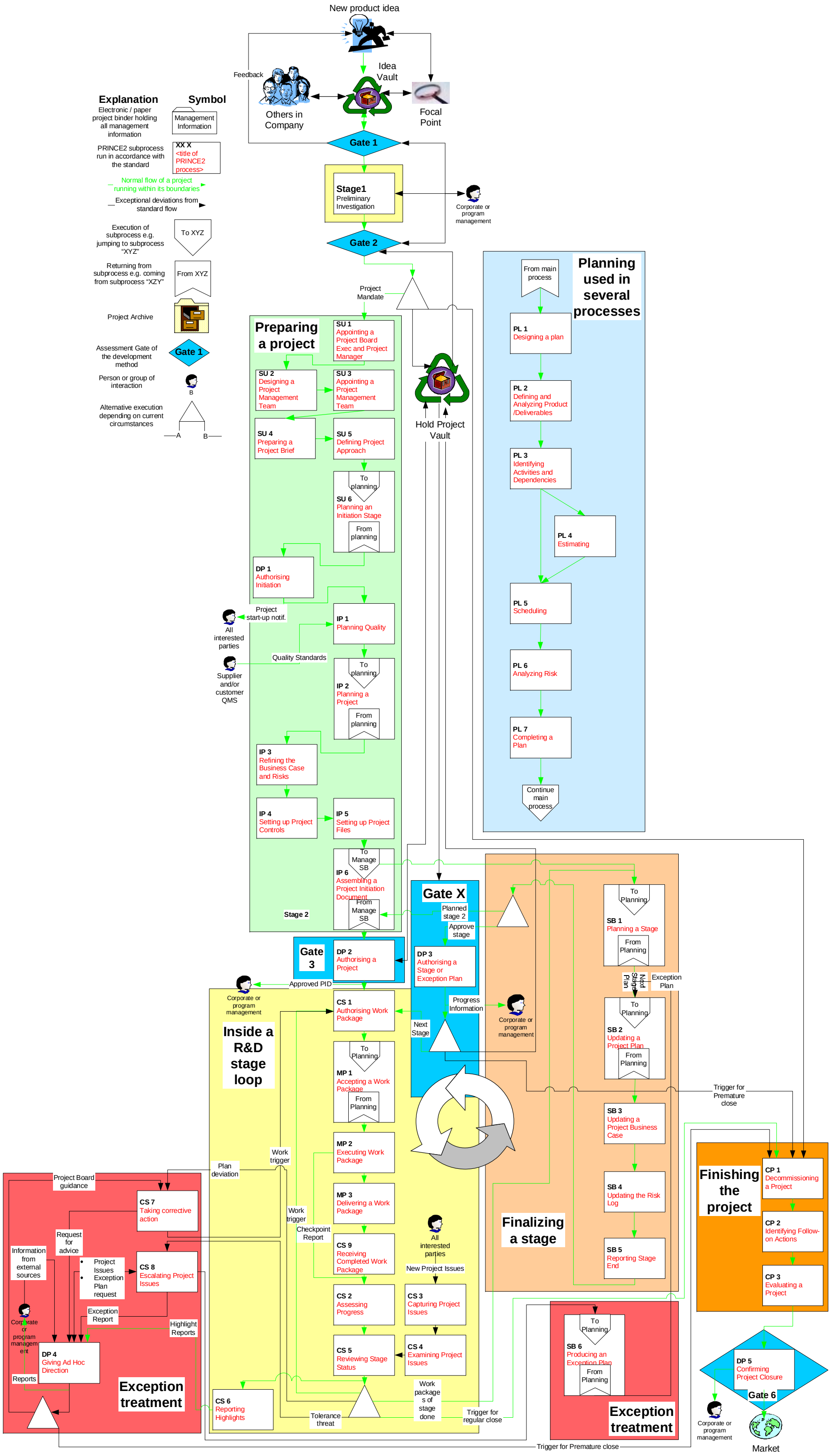


Figure 3.10.: Development method serialized

This is indicated by the arrow circle in figure 3.10. If exceptions or issues occur the additional steps of the exception treatment blocks might be used.

As can be seen from figure 3.10 the overall development process can be grouped into the two abstract phases of preparing a R&D project (everything until Gate 3, depicted in figure 3.12) and executing the R&D project loop. The details of the management steps to prepare the R&D project are described in section A.1. The R&D loop is of iterative nature in itself and is used in Stage 3, Stage 4 and Stage 5 as graphically presented in figure 3.10. A more detailed view of that loop is given in figure 3.15 and will be used later on. To prevent repetitive descriptions for these stages, the management aspects of the R&D loop are described in section A.2. All the steps, stages, deliverables, gates, criteria etc. of the method are described in the following sections, partly in reference to the loop description.

3.4.4. Stage 0: Idea capturing

The first crucial step in new product development is the product idea generation and capturing, the ideation. Without a proper management of this step only a reduced amount or no new products at all will be developed. The ideation can even be considered the first stage of the development process, but it is kind of before the process itself because it triggers the development process. Therefore it is named Stage 0.

For the considered business area new project ideas are typically related to device designs, circuit designs, new or improved manufacturing processes, new integration methodologies etc. These are only examples for the main area of product ideas. In principle all kinds of new product ideas (recall the definition of newness in section 3.1.1) can be handled with the same approach. This implies that even product improvement ideas, application of products to new areas etc. will be handled as a new product idea and evaluated, developed and commercialized along the same processes. The interactions in this stage and the flow of activities and events is presented in the left top corner of figure 3.11 and in figure 3.12.

To allow the systematic capture and evaluation of new product ideas the following four components are foreseen in the new development method:

Identifying the sources for new product ideas The sources for new ideas are typically the different R&D groups (process, design, modelling, etc.) as well as the marketing and sales (M&S) groups. These sources need to be identified and included in any idea fostering activity. Especially the input and feedback from the market via the M&S group is essential.

Establishing a focal point/person for idea generation The focal point is in charge to manage the idea generation. This needs to contain the stimulation of

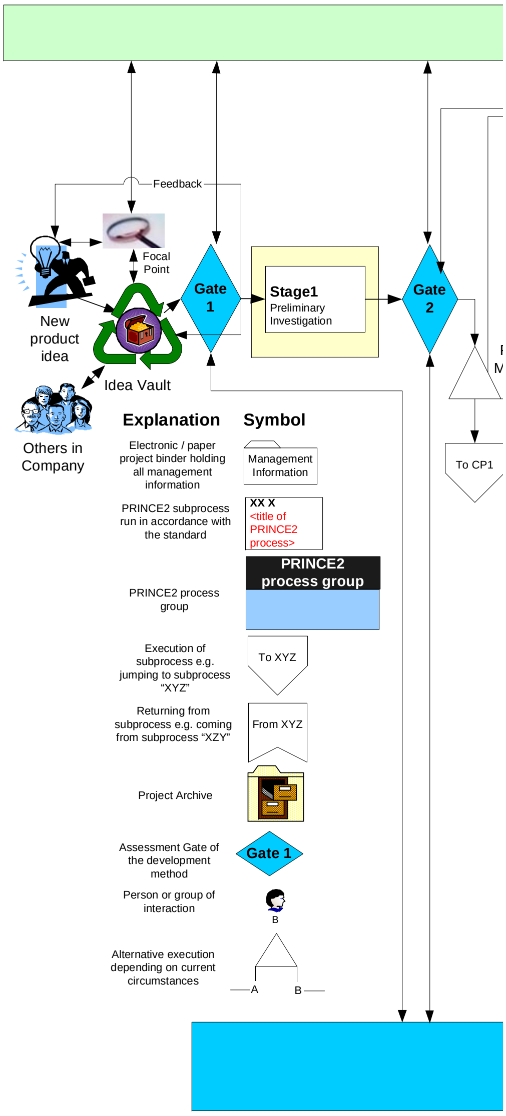

Feedback
Focal Point
New product idea
Idea Vault
Others in Company
Gate 1
Stage1
Preliminary Investigation
Gate 2
To CP1
Explanation
Symbol
Electronic / paper project binder holding all management information
Management Information
PRINCE2 subprocess run in accordance with the standard
XX X
<title of PRINCE2 process>
PRINCE2 process group
PRINCE2 process group
Execution of subprocess e.g. jumping to subprocess "XYZ"
To XYZ
Returning from subprocess e.g. coming from subprocess "XZY"
From XYZ
Project Archive
Assessment Gate of the development method
Gate 1
Person or group of interaction
B
Alternative execution depending on current circumstances
A
B

Figure 3.11.: Development method left part

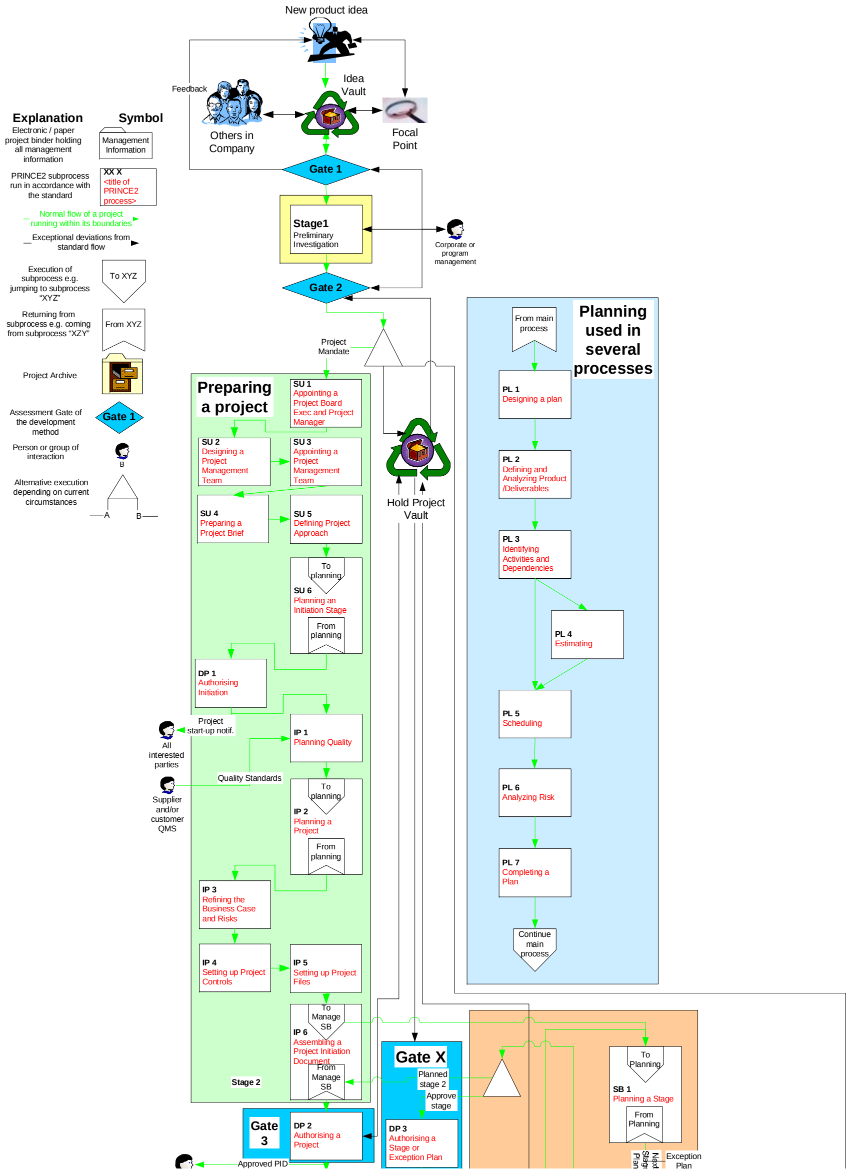

New product idea
Feedback
Idea Vault
Others in Company
Focal Point
Explanation
Symbol
Electronic / paper project binder holding all management information
Management Information
PRINCE2 subprocess run in accordance with the standard
XX X <title of PRINCE2 process>
Normal flow of a project running within its boundaries
Exceptional deviations from standard flow
Execution of subprocess e.g. jumping to subprocess "XYZ"
To XYZ
Returning from subprocess e.g. coming from subprocess "XZY"
From XYZ
Project Archive
Assessment Gate of the development method
Gate 1
Person or group of interaction
B
Alternative execution depending on current circumstances
A
B
Gate 1
Stage1
Preliminary Investigation
Corporate or program management
Gate 2
Project Mandate
Preparing a project
SU 1 Appointing a Project Board Exec and Project Manager
SU 2 Designing a Project Management Team
SU 3 Appointing a Project Management Team
SU 4 Preparing a Project Brief
SU 5 Defining Project Approach
To planning
SU 6 Planning an Initiation Stage
From planning
DP 1 Authorising Initiation
Project start-up notif.
All interested parties
IP 1 Planning Quality
Quality Standards
Supplier and/or customer QMS
To planning
IP 2 Planning a Project
From planning
IP 3 Refining the Business Case and Risks
IP 4 Setting up Project Controls
IP 5 Setting up Project Files
To Manage SB
IP 6 Assembling a Project Initiation Document
From Manage SB
Stage 2
Gate 3
DP 2 Authorising a Project
Approved PID
Hold Project Vault
Gate X
Planned stage 2
Approve stage
DP 3 Authorising a Stage or Exception Plan
Planning used in several processes
From main process
PL 1 Designing a plan
PL 2 Defining and Analyzing Product /Deliverables
PL 3 Identifying Activities and Dependencies
PL 4 Estimating
PL 5 Scheduling
PL 6 Analyzing Risk
PL 7 Completing a Plan
Continue main process
To Planning
SB 1 Planning a Stage
From Planning
Next Stage Plan
Exception Plan

Figure 3.12.: Serialized development method top part

the identified idea sources (see above) to generate new ideas and handling the new idea generation process. The focal point typically should be the CTO or COO, not in charge to generate ideas by himself, but to foster and handle the idea generation processes.

Grease the path The creativity itself can be increased via several means. As an example an idea kit could be given to M&S personal to easily capture ideas coming up from customer discussions. Another possibility is to set up brainstorming sessions, even with customers, to participate in trade shows and conferences, to foster that technical people visit customers etc. A summary of possibilities is given on page 111 of Coopers book [Coo98].

Idea capturing and handling system The most important features of an idea capture and handling system are to 1. allow easy capture/submission of ideas, potentially as an open discussion; 2. have a defined screening group applying a defined criteria catalog; 3. provide timely feedback to submitters in writing; 4. establish an idea vault for ″Kill″ and ″Hold″ ideas; 5. establish some award for successful ideas.
Most of these features can be fulfilled with setting up a Public Folder in a groupware solution e.g. MS Exchange. To ease and standardize the submission, an electronic groupware form is set up, asking for all necessary information about a product idea. The idea folder can be accessed for reading and writing by all employees to allow discussing about the ideas. For newly submitted ideas the provider and the focal point receives a reception notification. The folder is moderated by the focal point in the person of the CTO. Together with the group of gatekeepers of the first gate he discusses all ideas on a regular basis or on demand.

3.4.5. Gate 1: Idea screen

The first gate is a very gentle idea screen on criteria like strategic fit, market attractiveness, technical feasibility etc. The gate is executed by the gatekeepers consisting of one circuit design, one process development and one device design group leader, the CTO and a marketing representative. Depending on the product or improvement idea, additional decision makers can be consulted or included.

The first phase of the screening filters out the not viable new product ideas and is performed following the checklist presented in table 3.3. Only products and project ideas fulfilling all the criteria with either ″Yes″ or ″N/A″ will be assessed further with a scoring check. The failing ideas will be, together with their complete discussion, moved to the sub folder ″Killed″ of the discussion forum.

The scoring model check of the first gate is performed using the criteria checklist presented in table 3.4. If the averaged value for one criteria is below threshold or the overall average threshold is not met, the product idea will be put on ″Hold″, moved to the ″Hold″ sub folder in the groupware and will be reassessed later again. An additional possibility is to refer the idea back to the author for the addition of more information in case the information provided is not sufficient or not clearly expressed. In these cases the idea is moved to ″Hold″ as well. The gate does not only assess new product ideas. All ideas of status ″Hold″ in the vault are reassessed periodically as well and might be moved to a different category depending on the reevaluation results. This allows good ideas previously put on ″Hold″ to be revisited and pursued at a later point in time. This might happen because the market, the strategy or the technical feasibility has improved compared to the previous assessment. On the other hand, on hold ideas might be killed completely because the opportunities or strategies have changed. The results of the scoring check are used to rank and prioritize the different competing ideas. Ideas passing the criteria check but do not have a high enough priority will be sorted out into the idea vault in a ″Hold″ state and reassessed later as well. The winning ideas will continue into Stage 1. Dynamic reallocation of resources, as outlined in section 3.4.1, might be applicable as well. The interaction of the gate and the flow is presented in figure 3.11.

Criteria	**met Yes**	**met No**	**N/A**
Does the product fit to the overall business strategy?			
Is the market need preliminary confirmed?			
Might the potential product deliver unique customer benefit?			
Is there a reasonable chance of technical feasibility?			

Table 3.3.: Gate 1 Must-Meet criteria

Criteria	**Threshold**	**Rating (pt. 0-5)**
Degree of alignment with overall business strategy	3	
Degree of alignment with design portfolio strategy	3	
Degree of alignment with process portfolio strategy	3	
Degree of potential customer benefit	2	
Likeliness of technical feasibility	3	
Overall Threshold	17	

Table 3.4.: Gate 1 Should-Meet check-list

After the assessment of new ideas the provider receives a written notification about the judgement via the forms of the groupware. Additionally the idea, together with its complete discussion is moved to one of the sub folders categorized with the outcome of the assessment (Proceed development now, Hold, Kill). These folders are only writable to the focal point but can be read by all employees.

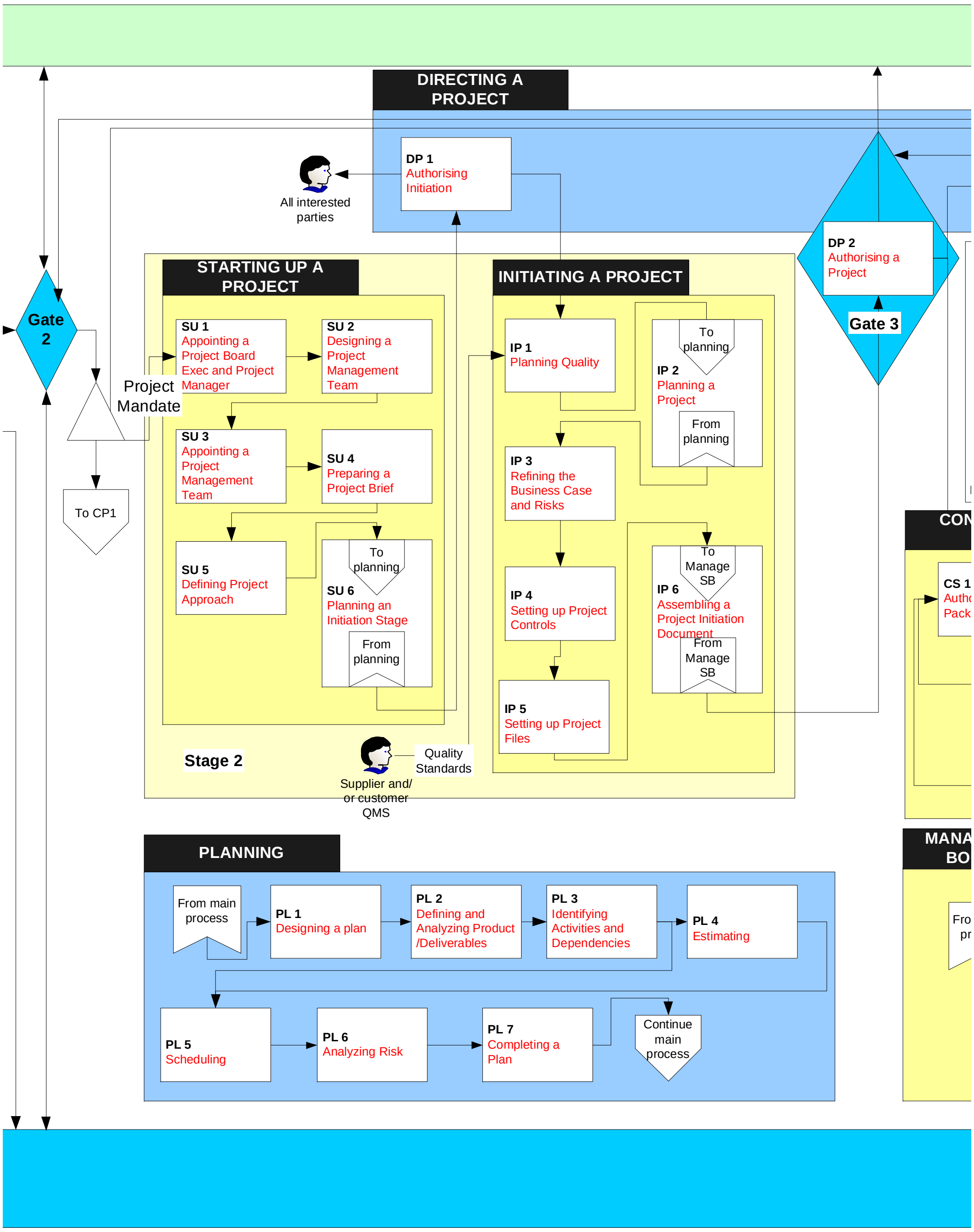

DIRECTING A PROJECT
DP 1 Authorising Initiation
All interested parties
DP 2 Authorising a Project
Gate 3
Gate 2
STARTING UP A PROJECT
SU 1 Appointing a Project Board Exec and Project Manager
SU 2 Designing a Project Management Team
SU 3 Appointing a Project Management Team
SU 4 Preparing a Project Brief
SU 5 Defining Project Approach
SU 6 Planning an Initiation Stage
To planning
From planning
Project Mandate
To CP1
INITIATING A PROJECT
IP 1 Planning Quality
IP 2 Planning a Project
To planning
From planning
IP 3 Refining the Business Case and Risks
IP 4 Setting up Project Controls
IP 5 Setting up Project Files
IP 6 Assembling a Project Initiation Document
To Manage SB
From Manage SB
Stage 2
Quality Standards
Supplier and/ or customer QMS
CON
CS 1 Autho Pack
MANA BO
Fro pr
PLANNING
From main process
PL 1 Designing a plan
PL 2 Defining and Analyzing Product /Deliverables
PL 3 Identifying Activities and Dependencies
PL 4 Estimating
PL 5 Scheduling
PL 6 Analyzing Risk
PL 7 Completing a Plan
Continue main process

Figure 3.13.: Development method middle part

3.4.6. Stage 1: Preliminary investigation

Stage 1 briefly determines the project's technical, financial, legal/patent and marketplace merits and evaluates in more detail whether the project strategically fits. Only about ten to twenty person-days are spent in, usually, less than a month time to perform a preliminary investigation of the product idea. Therefore one part-time resource from each of the relevant technical groups (depending on the idea classification e.g. design and process group) will be assigned to the project. Additionally one part-timer from the M&S group, from the legal/patent group and from finance group will be assigned to the investigation. The embedding of the stage into the overall method is presented in figure 3.13.

Deliverables of Stage 1

Stage 1 delivers several initial reports described below and listed in table 3.5. They contain only very preliminary data because this phase of the project is very brief and is executed with a small amount of resources. The deliverables are updated and extended or used as input in later phases of the project. They contain the following information:

Preliminary market assessment report This report contains information about the first scoping of the market consisting of market existence, probable market size and market acceptance and how the idea could be placed into the market. The market opportunity window and competition should be briefly investigated as well. Since the resources are limited the collection of this data is mostly desk and Internet research.

Preliminary technical feasibility report This report contains a first idea about realization possibilities, probabilities and potential development costs, times and risks. A proposal for a potential route forward should be contained as well. For the assessment a literature study and mostly internal, may be some external, technical expertise and resources are used. These efforts can be supported by the system described in chapter 6. It captures information from previous development efforts about materials, process steps and flows, assessment results, details about the development sequence, simulation results etc. in a database. The system provides an overview about existing knowledge enabling easier and faster assessments and provides information for quantifying ideas. The statistical information captured in the system provides a profound planning and scheduling basis. The report is supposed to collect information about existing processes or process steps, integration schemes, designs and so forth usable for the new product idea.

Preliminary prospects report The Preliminary prospects report contains an initial assessment on the potential product prospects. It takes the very rough estimates of potential sales (received from market assessment), costs, required investment and risk (from technical feasibility assessment) into account. Therefore it builds a stepping stone for the creation of a business case in the later stages.

Preliminary legal assessment report This report contains results of an initial legal assessment covering laws, regulations and patents. Because the targeted environment develops IP products, an initial patent database check via the Internet on the product idea is essentials. The evaluation has to check whether there are blocking or competing patents in place and whether there are any negative legal implications developing a product based on the new idea. The early inclusion of such an initial legal/patent assessment is supported by e.g. [Wik05g].

Draft Project Mandate The information of the above three reports will be summarised in the draft Project Mandate. The summaries of the other preliminary deliverables of the stage have to be added to the corresponding chapters in the Project Mandate template. Such a template is provided with the method and is described in more detail in section A.3.1. Additionally references to the reports will be added to the mandate as well. In the case of a positive assessment during Gate 2, the draft project mandate will be extended/completed and agreed during the gate execution and used to initiate the Stage 2.

Initialization Stage plan The Initialization stage plan lays out the schedule for the developments of the project initialization stage. This schedule needs to contain all tasks necessary to setup up the project development environment including the electronic project binders, etc. The generation of the plan follows the planning guidelines summarised in section A.2.

Request for authorization to proceed This request documents, on completion, that the project is authorized to proceed into the next stage and grants all required resources.

Deliverable	Compulsory	Optional
Preliminary market assessment report	X	
Preliminary technical feasibility report	X	
Preliminary prospects report	X	
Preliminary legal assessment report	X	
Draft Project Mandate(filled template)	X	
Project Initialization Stage plan(filled template)	X	
Req. for authorization to proceed	X	

Table 3.5.: Stage 1 deliverables

3.4.7. Gate 2: Detailed screen

The Gate 2 is basically a repetition of the Gate 1 assessment but in the light of the additional information gathered during Stage 1. Therefore the assessment is a bit more thorough. It uses a must-meet list, presented in table 3.7, and a scoring model, presented in table 3.8. Both catalogues are supplemented with additional criteria, compared to Gate 1, to assess the additional available information.

The decision making during the gate assessment is similar to Gate 1. The Must-Meet criteria list is applied to the project and failing projects are "Killed". Projects taking this hurdle will be assessed with the Should-Meet check-list presented in table 3.8. The selection and prioritisation of the projects is executed in a similar manner as during Gate 1. Projects scoring too low will be put on "Hold" and reassessed during a later Gate 2 assessment again. Therefore the gate assessment includes all projects of the Hold project vault from this development stage. Additionally outstanding projects or product ideas might cause reallocation of projects in different stages as well, as mentioned in section 3.4.1. For the winning projects the draft project mandate delivered to the gate will be finalized during the assessment session, filling, or at least deciding on, the remaining open items in the mandate. Shortly after the gate assessment session the finalized project mandate is used to initiate the Project Initialization Stage 2 via triggering the decided project executive and project manager with the project mandate as input. Additionally the delivered stage plan for the initialization stage is approved as is or amended and approved as an execution basis for the next stage.

Inputs	Activities	Outputs
Prel. market assessment report Prel. technical feasibility report Prel. prospects report Prel. legal assessment report Draft Project Mandate Initialization Stage plan Req. for authorization to proceed	Authorising a Project Initialization	Approved Initialization Stage Plan Approved Project Mandate Authorization to proceed

Table 3.6.: Gate 2 activities with inputs and outputs

Criteria	met Yes	met No	N/A
Does the product fit to the overall business strategy?			
Does the product fit to all business groups strategies?			
Is the market need preliminary confirmed?			
Is the addressed market of attractive size and is it growing?			
Might the potential product deliver unique customer benefit?			
Is there a reasonable chance of technical feasibility?			
Are there only minor or no legal/IP issues?			

Table 3.7.: Gate 2 Must-Meet criteria

Criteria	Threshold	Rating (pt. 0-5)
Degree of alignment with overall business strategy	3	
Degree of alignment with design portfolio strategy	3	
Degree of alignment with process portfolio strategy	3	
Degree of potential customer benefit	2	
Degree of potential customer need	2	
Degree of trade appeal and fit	2	
Degree of market size and growth appeal	2	
Likeliness of technical feasibility	3	
Appeal for patent portfolio	3	
Overall Threshold	25	

Table 3.8.: Gate 2 Should-Meet check-list

3.4.8. Stage 2: Project Initialization

During Stage 2 the project is initialized according to the PRINCE2 method. Some additions for the combined method have been added as outlined below. The Stage only slightly progresses the technical, marketing, financial and legal status of the project. It is more a management and infrastructure stage, typically performed within a short period of time (one to two weeks).

The main purposes of the stage are to perform the project initialization by appointing the project responsible persons, setting up the project infrastructure, developing the project brief and project initiation document out of the project mandate, setting up an initial business case and develop the different plans. The technical progress is limited to defining the vision for the final product, the outcome of the development project, within the Product Brief. The importance of this vision is described in more detail in section 4.14. For that the project mandate and the preliminary feasibility reports produced during the previous stage are used. The distinction between the project and the product brief is that the project brief is more concerned about the execution of the development project while the product brief tries to capture an early vision of the final product. Therefore the product brief builds an elementary block of the ”Vision” brick of the Engineering Framework described in section 4.14. More details about the product brief can be found in section A.3.4.

The sequence of actions follows the PRINCE2 standard for the process groups ”Starting up project” and ”Initiating a project” and is described in more detail in section A.1. Additions and extensions of the PRINCE2 standard are contained in the description as well. They are given by further explanations or as additional inputs or outputs e.g. the addition of setting up an electronic project binder. For the creation of these project binders a set of project create scripts have been implemented as documented in section 4.5. The created binder and project repository contains a full set of project management templates in MS Word, MS Excel or MS Project format tailored to the needs of the product development method. For that new templates have been created or the contents of the templates proposed in [GC02] and [Ben02] is adapted and extended. The set contains templates for all aspects and project management documentations.

Deliverables of Stage 2

Stage 2 delivers, apart from all the project management artefacts, intermediate results and project infrastructure documentation (e.g. Project Brief, Risk and Issues Logs, Project Plan and Project Quality Plan, updated Business Case) the deliverables listed in table 3.9. The high quality preparations of these deliverables is a key assessment criteria in the proceeding Gate 3.

Feasibility Stage plan The Feasibility stage plan lays out the schedule for the developments of the product definition, business case, product requirements and product positioning strategy, legal research report and risk&issues report. It schedules the different work packages and activities to develop these deliverables during the next stage. The generation of the plan follows the planning guidelines summarised in section A.2.

Deliverable	Compulsory	Optional
Initiation Stage Plan(filled template)	X	
Project Brief(filled template)	X	
Product Brief(filled template)	X	
Draft Project Initiation Document(filled template)	X	
Feasibility Stage plan(filled template)	X	
Stage end report(filled template)	X	
Req. for authorization to proceed	X	

Table 3.9.: Stage 2 deliverables

3.4.9. Gate 3: Project gate

Gate 3 assesses the project with the same criteria catalogues as Gate 2, shown in table 3.7 and 3.8. It only reverifies whether the evaluation done at Gate 2 is still valid. The gate has been introduced to reevaluate the project again before the real developments and the spending into the project occurs.

The gate only checks whether the overall situation has change since Gate 2 and additionally verifies the proper and quality execution of the project initialization phase. Therefore it additionally checks and approves the creation of the Initiation Stage Plan (although already passed but checking for quality aspects), Project Brief, Product Brief, Draft Project Initialization Document (PID) and the Feasibility Stage plan. Focus is also put on the proper preparation of the overall project and quality plans. The gate is only executed between the project manager and project executive without higher level management involvement so that it can be considered an intermediate gate. Only if discrepancies between the two gates concerning the catalogue evaluation are discovered or a disagreement between the executive and the project manager is detected, senior management becomes involved into the gate execution. The outcomes of the gate are similar to Gate 2. The project might be put on "Hold", although very unlikely, or the Feasibility Stage plan and the PID becomes approved. Recycles might occur in cases of incomplete or too low quality execution of the Stage 2. Table 3.10 shows the PRINCE2 process executed to perform the gate assessment together with the gates inputs and outputs.

Inputs	Activities	Outputs
Initiation Stage Plan Project Brief Product Brief Draft PID Feasibility Stage plan Stage end report Req. for authorization to proceed	DP2 Authorising a Project	Approved Feasibility Stage Plan Approved PID Authorization to proceed

Table 3.10.: Gate 3 activities with inputs and outputs

3.4.10. Stage 3: Feasibility study

Stage 3 is where a more detailed business case for a product idea is constructed and a detailed feasibility analysis is performed. It also can be considered to be the critical homework stage where the technical, marketing, legal and financial feasibility is worked out and the product is defined in detail. Its proceeding gate builds the door to the products prototype development stage where the heavy spending into the product starts. The Stage 3 consumes about ten to twenty percent of the overall projects development budget and therefore marks a considerable step forward on the way to a marketable product. Stage 3 is characterized by requiring all involved business groups and subjects performing their homework. The outcomes must show that the product is feasible and that there is good evidence that the spending into the project will pay-off. Because this stage already consumes a considerable amount of resources and therefore requires a considerable amount of coordination, the previous stage has set up the project management framework and performed the initializing steps.

The stage performs several activities, grouped into a set of work packages. It produces the deliverables for the next gate using an multidisciplinary team. These deliverables are summarised in table 3.11 and are described below. Although the activities can be grouped into the different technical and administrational areas to be covered, the developments are performed in close cooperation between the multidisciplinary people. The stage answers questions like "What is the product and who will it be sold to?" and includes a specification of the target market. It generates a description of the product concept and the

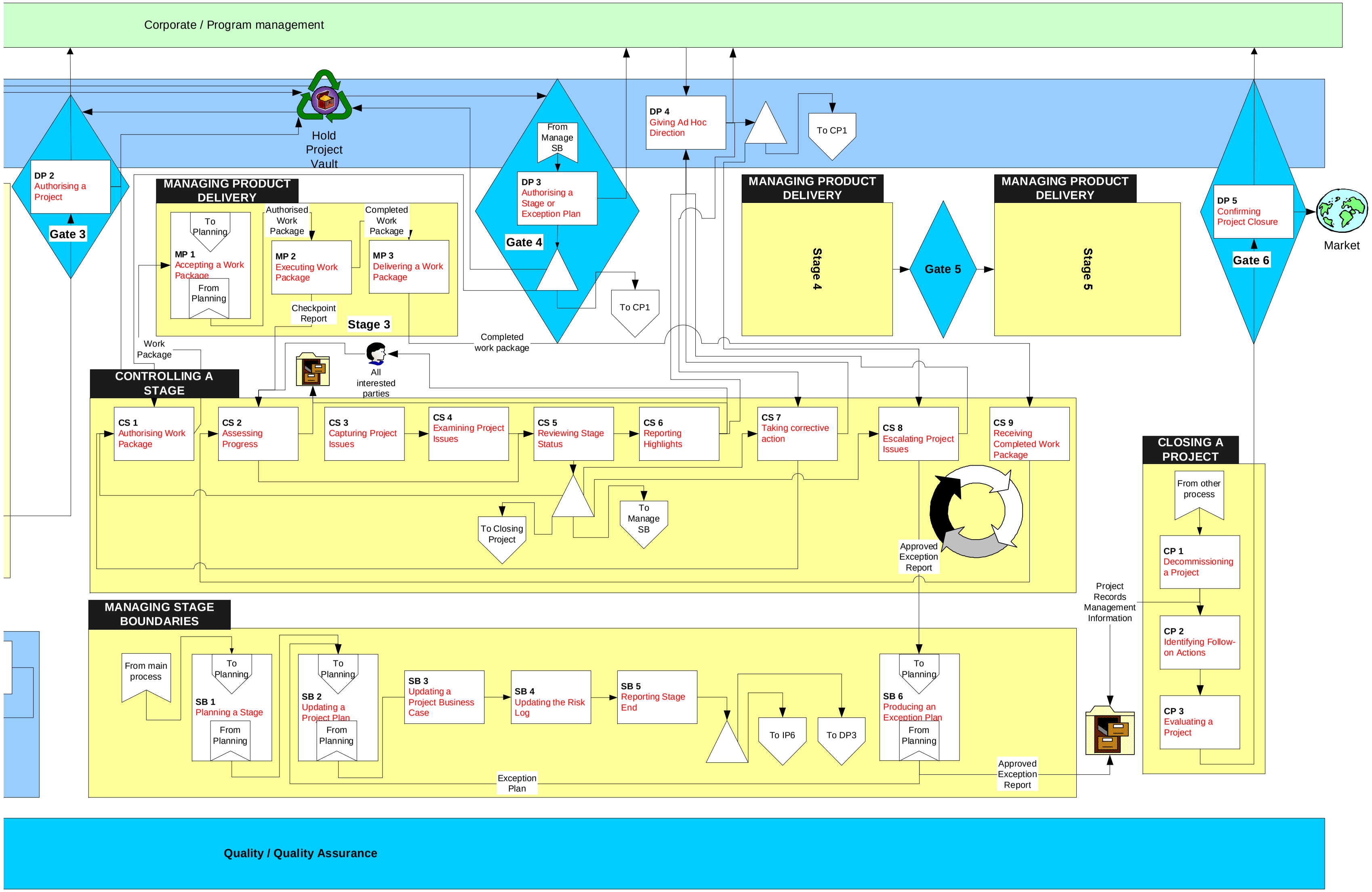

Corporate / Program management
Hold Project Vault
DP 2 Authorising a Project
Gate 3
MANAGING PRODUCT DELIVERY
To Planning
MP 1 Accepting a Work Package
From Planning
Authorised Work Package
MP 2 Executing Work Package
Completed Work Package
MP 3 Delivering a Work Package
Checkpoint Report
Stage 3
From Manage SB
DP 3 Authorising a Stage or Exception Plan
Gate 4
To CP1
Completed work package
DP 4 Giving Ad Hoc Direction
To CP1
MANAGING PRODUCT DELIVERY
Stage 4
Gate 5
MANAGING PRODUCT DELIVERY
Stage 5
DP 5 Confirming Project Closure
Gate 6
Market
Work Package
All interested parties
CONTROLLING A STAGE
CS 1 Authorising Work Package
CS 2 Assessing Progress
CS 3 Capturing Project Issues
CS 4 Examining Project Issues
CS 5 Reviewing Stage Status
CS 6 Reporting Highlights
CS 7 Taking corrective action
CS 8 Escalating Project Issues
CS 9 Receiving Completed Work Package
To Closing Project
To Manage SB
Approved Exception Report
CLOSING A PROJECT
From other process
CP 1 Decommissioning a Project
CP 2 Identifying Follow-on Actions
CP 3 Evaluating a Project
Project Records Management Information
MANAGING STAGE BOUNDARIES
From main process
To Planning
SB 1 Planning a Stage
From Planning
To Planning
SB 2 Updating a Project Plan
From Planning
SB 3 Updating a Project Business Case
SB 4 Updating the Risk Log
SB 5 Reporting Stage End
To IP6
To DP3
To Planning
SB 6 Producing an Exception Plan
From Planning
Exception Plan
Approved Exception Report
Quality / Quality Assurance

Figure 3.14.: Development method right part showing the R&D loop

delivered benefits, a delineation of the positioning strategy, a list of product features, attributes, requirements and high-level specifications. Other questions to be answered are ”Why to invest into this project?” where a thorough project justification (strategic, financial, risk) needs to be worked out and ”How will it be undertaken; when, by whom, and how much will it cost?” This might require an update of the project development plan handled by the Finalizing a Stage process steps.

All the technical, financial, legal and M&S activities require coordination and management by the PRINCE2 processes and sub-processes. These management activities are executed using the process groups ”Inside a R&D stage”, ”Finalizing a Stage” and ”Gate assessment” as detailed in section A.2. The loop inside the single stage and its process steps are depicted in figure 3.14 and 3.15.

Deliverables of Stage 3

Product definition In close collaboration with the M&S oriented activities detailed technical assessments are performed. As a result the of these assessments the product definition is delivered. Here initial lab work e.g. material selection, process step selection etc., product modelling and simulation, test designs etc. will be performed to evaluate the technical feasibility in more detail.

These activities can be supported by the system described in chapter 6 by providing a profound basis of knowledge and supporting DFM assessments, simulation and experimental verification. Even first prototypes with existing equipments, process steps and flows, materials, parts, designs and techniques might be performed to get a better feeling for the further development efforts. Feeding the results of these efforts back into the Design and Tracking Systems enables reproducibility in the next stages and extends the knowledge base. For the technical feasibility check, the rule of thumb of spending about ten percent of the development costs in this phase should be watched, otherwise the majority of the development effort will already be done in this stage.

The technical activities feed directly into the M&S activities trying to evaluate the market potential. Additionally they are required for the assessment concerning the manufacturability at a customer sight, the necessary customer investments and so on. The M&S activities feed back into the technical efforts to allow amendments to the concept and to better fit the product characteristics to the customer needs. All these efforts are bundled into the write up of the product definition. This definition can have varying contents depending on the application domain of the product. A more detailed description of the product definition is given in section A.3.5.

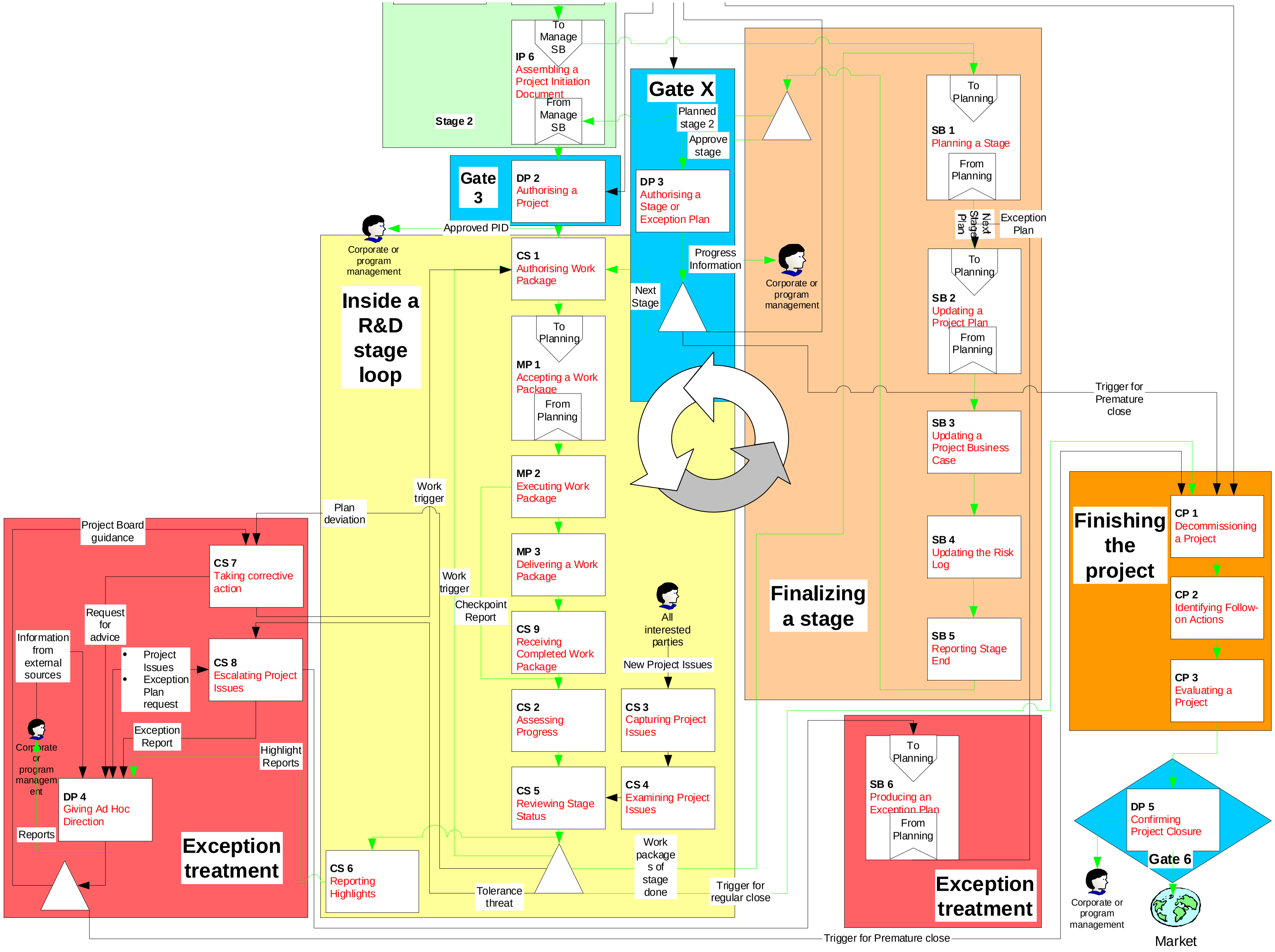


Figure 3.15.: Serialized method bottom part

Business case Besides the technical activities, a more detailed financial analysis justifying the investment into the project are performed. They have the target to deliver a more detailed business case which becomes even more detailed step by step in the proceeding stages as the knowledge about the product, its market value, etc. increases. An important component requiring consideration are the products and projects risks and costs especially from the legal and IPR area. The financial analysis typically uses discounted cash flow methods, break-even analysis or return of investment calculation approaches. They are executed by the financial people assigned to the project.

Requirements & Positioning strategy The M&S responsible persons have to, at least, perform a user needs-and-wants study, evaluating the customers preferences, likes, dislikes, performance requirements and so on and summarise them in the requirements specification deliverable. A template for such a specification is described in more detail in section A.3.3 and belongs to the electronic project binder. Additionally a competitor screening needs to be done evaluating on their strengths, weaknesses, business performance, etc. A value-in-use analysis enables a comparison between the predicted own product characteristics with the alternative market solutions. Additionally it allows determining the cost-benefit ratio from the customer's perspective. Where applicable, first paper prototype presentations with customers might be performed allowing to get feedback from potential customers about the envisioned product characteristics. The result of these activities need to be summarised in the Positioning strategy report which has to be delivered to the next gate.

Legal Research Report Considering the results of the technical feasibility and basing it on the preliminary legal investigation from Stage 1, a more detailed legal, regulatory and patent assessment needs to be performed. Here comparable approaches and patents need to be searched and evaluated to prevent later legal and IP issues. After fixing the approach and being sure about the feasibility of the product idea, first steps towards protecting the new product might be taken. This includes writing up new patent applications, registering new trademarks or copyrights, etc. All activities and results need to be documented in the legal research report.

Risks&Issues Report An explicit risk and issue report is required for the next gate because the next gate needs to take serious decisions and authorize heavy spending into the project and the product idea. The report needs to list and quantify all open issues and risks contained in the issue and risk log to address them explicitly. For all issues and risks a discussion about likeliness, impacts, workarounds etc. has to be worked out.

Prototype dev. Stage plan The prototype development stage plan manifests the Product Breakdown Structure(PBS; defined is section A.2), the deliverable descriptions and the deliverable dependencies for the next stage. It schedules the different work packages and activities to develop the product prototype(s). It plans the execution of the different market sampling, prototype test, refinement of the business case and resolution of the legal issues. For that it follows the planning guidelines summarised in section A.2.

Deliverable	Compulsory	Optional
Product definition(filled template)	X	
Business Case(filled template)	X	
Requirements Specification(filled template)	X	
Positioning Strategy(filled template)	X	
Legal Research Report	X	
Risks&Issues Report(filled template)	X	
Prototype dev. stage plan(filled template)	X	
Stage end report(filled template)	X	
Req. for authorization to proceed	X	

Table 3.11.: Stage 3 deliverables

3.4.11. Gate 4: Prototype-Development decision

This Gate 4 can be considered as the ”Go-to development gate”. Up to now only initial checks and feasibility tests have been performed, in the Stage 4 a prototype of the product has to be developed. Because of that the heavy spending into the product idea starts with the following stage. Therefore this gate needs to be executed very thoroughly with higher management involvement to reach sign-off on the product definition, the business case, the development and the marketing plans. Additionally agreement on the commitment of the required resources (capital, personal, etc.) needs to be achieved. The priority shifts are reflected in the Must-meet (see table 3.13) and in the Should-meet (see table 3.14) criteria catalogues.

The outcomes of the gate are similar to previous gates. The project might be put on ”Hold” because of different prioritisation, Stage 3 might be repeated because of the low quality or missing of deliverables or the project might be killed altogether because of various reasons(strategic fit, risks, issues, etc.). In case of a positive assessment, the Prototype Development Stage plan, Business Case and the other inputs become approved and the project can proceed. Table 3.12 shows the PRINCE2 process executed to perform the gate assessment together with the gates inputs and outputs.

Inputs	Activities	Outputs
Product definition Business Case Requirements Specification Positioning Strategy Legal Research Report Risks&Issues Report Prototype dev. Stage plan Stage end report	DP3 Authorising a Stage or Exception Plan	Approved Prototype dev. Stage Plan Approved Product definition Approved Business Case Authorization to proceed

Table 3.12.: Gate 4 activities with inputs and outputs

Criteria	met Yes	met No	N/A
Product still fits to the business strategy			
Product still fits into design portfolio			
Product still fits to the process portfolio			
Market need and profitable size confirmed			
Market requirements clearly defined			
Positioning strategy clearly defined			
Product Definition clear and agreed			
Business Case gives profitable outlook			
No major legal risks or issues			
Technical Risks and Issues are acceptable			

Table 3.13.: Gate 4 Must-Meet criteria

Criteria	Threshold	Rating (pt. 0-5)
Degree of alignment with overall business strategy	3	
Degree of alignment with design portfolio strategy	3	
Degree of alignment with process portfolio strategy	3	
Degree of market size and growth appeal	3	
Quality of requirements	2	
Appeal of positioning strategy	3	
Degree of potential customer benefit	2	
Degree of customer need	2	
Degree of trade appeal and fit	2	
Degree of newness of the product	2	
Clearness of Product definition	3	
New Process knowledge captured in Design and Tracking System	4	
Profitability according to business case	3	
Level of risk (low risks→high # of points)	4	
Level of legal risks and issues (low risks→high # of points)	4	
Positive projects self-assessment	3	
High quality prototype dev. stage plan	3	
Overall Threshold	52	

Table 3.14.: Gate 4 Should-Meet check-list

3.4.12. Stage 4: Prototype development

Stage 4 executes the prototype development stage plan as decided in Gate 4. The stage delivers one or several product prototypes while these can consist out of several components. These components are verified to a level that in-house lab-testing and initial testing and quality assurance measures have been performed. In certain cases it might be possible to perform preliminary customer tests as well. For the prototype development efforts it should be noted that postponement and slippage is only allowed within the agreed project thresholds. Otherwise the project manager has to issue an exception report to the project steering committee which might assign additional resource or request additional effort. Serious project delays should be prevented in any circumstances because they might surrender the business case or the whole project useless. All threshold overshoots need to be treated according to the mechanisms document in section A.2. Because the execution of this stage typically requires a longer period of time, intermediate project or project management audits might be performed to ensure the high quality execution of the developments.

Deliverables of Stage 4

Product prototype(s) During the prototype development stage one or more product prototypes will be developed. Such a prototype typically consists of several components representing the generated knowledge.

For the development of the process IP the process development methodology highlighted in section 3.5 can be used until the product demonstrator is achieved. The later steps are part of the following development stage. These efforts need to be supported and tracked with a systematic approach for which the system described in chapter 6 has been developed. It is used to perform DFM tests in software to prevent expensive and time consuming experimentation. Verification by process simulations is supported as well and it can be used support and capture the experimental verification with all experiments, lots, wafers, artefacts and so forth. All this information is captured in a database containing the materials, process steps and flows used to generate the prototypes as well as all the assessment results, details about the development sequence, simulation results etc. The database created for these prototypes can be selectively exported from the system and can be used as a vehicle to transfer the knowledge about process IP to customers. Additionally it is used to support the next stages and future developments.

Further example components of a prototype product, besides the above database, are a process blue book together with prototype wafers processed with that blue book, test designs with potentially fabricated test chips, integration manuals with feasible and tested device integration schemes, etc.

Marketing&Sales prototype report The M&S prototype report contains the results from constant market sampling and initial, as far as possible, customer prototype test. The market sampling needs to carefully observe the marketplace whether e.g. the market needs or the competitive situation concerning the project changes. In cases of major changes a report has to be generated, using the Project Issue template, making the project aware of the new facts. These have to be taken carefully into account (see process CS3 in section A.2) and might require further escalation using the exception treatment steps of the method (see section A.2).

Another potential outcome of the M&S market sampling are shifting requirements. In cases of shifted requirements M&S might issue a Change Request to the project, using the CR template (section A.3.6) + procedure (section 4.3.3), to get the new requirements considered and the implications evaluated. Apart from these sampling and intervention activities the M&S responsible person(s) might initiate and discuss initial customer test with the prototypes of the different stages. Results of these activities are fed back into the project and summarised into the M&S prototype report for the next gate. This report summarises all M&S activities and needs to cover at least the identified issues, the change request and the results of the customer prototype tests.

Legal Resolution Report Besides the other activities of this stage, the existing legal issues (law, regulatory, patent, etc.) need to be resolved in this stage. The resolution can have various flavours ranging from adding additional patents to protect the new idea, to resolve issues with infringed patents (buying license, other agreement), clarifying other legal issues etc. The outcome might be positive or negative but the aim must be to finalize the resolution of all legal issues. Whatever steps are necessary to resolve legal issues need to be taken in this stage. The results of this resolution process need to be documented in the Legal Resolution Report, especially not solved issues or deadlock situations need to be addressed.

Updated Business case A further deliverable is an updated business case. As the knowledge about the project and the final product grows, initial assumptions contained in the business case can be substituted by more well-founded data about the market, legal situation etc. Therefore the foundation data for the business case becomes more solid and the business case needs to reflect this fact. The updated business case becomes reassessed in the proceeding gate to complement the projects and products picture.

Development Stage plan In the following stage the knowledge generation for the different IP components continues. Of special importance is the prototype characterization according to semiconductor test procedures, the widening of the process windows for the different manufacturing process steps and the test-chip assessments of the prototype test-chips of the current stage. All deliverables of the next stage need to be considered in the Product Breakdown Structure(PBS; defined is section A.2), the deliverable descriptions and the deliverable dependencies. The plan needs to schedule the different work packages and activities to develop the final product deliverables ranging from technical, legal, financial and M&S deliverables. The generation follows the guidelines summarised in section A.2.

Deliverable	Compulsory	Optional
Product prototype(s)	X	
M&S prototype report	X	
Updated Business Case (filled template)	X	
Legal Resolution Report	X	
Development stage plan(filled template)	X	
Stage end report(filled template)	X	
Req. for authorization to proceed	X	
Audit protocols		X

Table 3.15.: Stage 4 deliverables

3.4.13. Gate 5: Development decision

After the prototype development stage the focus of the gate assessments shifts. From the question whether to invest into the project the focus is now put on questions like ”What is the progress?”, ”Is the project on track?”. Additionally the completeness, the ”positiveness” and the quality of the deliverables moves into the focus of the assessments. Although the focus shifts, still the technical, financial and strategic attractiveness of the product is reviewed but with reduced priority. These shifts are represented by a change in the Must-meet (see table 3.17) and Should-meet (see table 3.18) criteria catalogues.

The outcomes of the gate are similar to previous gates. The project might be put on ”Hold” because of changed priorities or Stage 4 might become repeated because of to low quality or missing deliverables. The project might be killed altogether as well because of various reasons(strategic fit, risks, issues, etc.), although the likeliness is reduced because the already high investments into the project. In case of a positive assessment, the Development Stage plan, the Updated Business Case and the other inputs become approved and the next stage becomes authorized.

Table 3.16 shows the PRINCE2 process executed to perform the gate assessment together with the gates inputs and outputs.

Inputs	Activities	Outputs
Product prototype(s) M&S prototype report Updated Business Case Legal Resolution Report Development stage plan Stage end report Req. for authorization to proceed Audit protocols	DP3 Authorising a Stage or Exception Plan	Approved Development Stage Plan Approved prototype(s) Approved Business Case Authorization to proceed

Table 3.16.: Gate 5 activities with inputs and outputs

Criteria	met Yes	met No	N/A
Product still fits to the business strategy			
Market outlook(size, need, positioning) still suitable			
Business Case gives profitable outlook			
Prototype(s) of good quality			
Legal risks or issues resolved			

Table 3.17.: Gate 5 Must-Meet criteria

Criteria	Threshold	Rating (pt. 0-5)
Degree of alignment with strategies	3	
Quality of prototype(s)	4	
Degree of market outlook appeal	4	
Positive M&S prototype report	4	
New Process knowledge captured in Design and Tracking System	4	
Profitability according to business case	4	
Level of legal risks and issues (low risks→high # of points)	4	
Audit reports positiveness	3	
Positive projects self-assessment	3	
High quality development stage plan	3	
Overall Threshold	38	

Table 3.18.: Gate 5 Should-Meet check-list

3.4.14. Stage 5: Product development

Stage 5 develops all components (internal and external) for all involved areas of the final product. This ranges from M&S plans (roll-out, distribution, support, etc.) and materials (product brochures, other collateral, etc.) to contract templates and an updated business case. The final technical deliverables contain elements like process manuals, characterization and reliability data, mask designs, etc. For the progression the boundary constraints of the stage execution as for the prototype developments apply. This includes the remarks made about project delays and exception treatment as well as for intermediate project and project management audits.

Deliverables

Technical product components During the development stage all required technical product components will be developed. For an combined process IP and design IP product the list of components can be grouped into Process IP Modules, Performance Indicators, Instantiation IP Modules and Design IP modules which are explained in a bit more detail below.

The Process IP modules will be developed along the methodology highlighted in section 3.5. The prototype development stage created the single product demonstrator solution. The development stage extends the solution into the solution corridor to widen the process windows. It basically performs the rolling backwards phase described in section **??**. To widen the process windows, virtual and real experiments need to be performed. These again can be supported and need to be captured by the system described in chapter 6. Essential process IP modules delivered by these activities are the detailed process description with the full recipe and process windows and the filled/exported database of the Process Design and Tracking Environment.

Performance indicators of the manufactured prototype devices can be derived from diverse assessments and tests. They are required to prove the maturity and the thoroughness of the developments to the customers. They only can be indicators because similar devices manufactured in a different environment can have slightly different properties. The required indicators must supply information about the characterizations of the process window for process and device variations (geometry or device type); wafer yield data potentially collected from different environments and setups; data about environmental and reliability test results.

The Instantiation IP modules represents knowledge on how devices fabricated by the Process IP modules can be integrated into the customers manufacturing processes. These include the definition of several generic integration schemes (potential for front end of line and back end of line), a generic process step description of the process IP, a customer survey to evaluate the customers fab capabilities and limitations and required qualification vehicles. The qualification vehicles are to support the different verification steps during the process porting and integration efforts. They cover material qualification, device qualification and so forth. Additionally diverse procedures, templates, plans, checklists and guidelines for the execution of the porting and integration project have to be set up.

The Design IP modules will consist of device and cell designs for the qualification vehicles and designs using the device(s) manufactured with the process IP. This could be e.g. a memory instances using a MEMS based memory cell. Additionally the design IP consists of design rule manuals and design kits allowing the usage of the design IP modules within customer instances and the customers design flow. The design kits will consist of e.g. simulation modules, cell libraries, DRC, LVS and PEX results, schematic symbols. etc. During the development of these deliverables a detailed assessment of the test chips developed and manufactured during the prototype development stage will be performed. The results will be used for the generation of the final deliverables.

M&S product components During the development stage the M&S activities are continued and even intensified. They have to plan, and partly already start, the roll-out, distribution, support, etc. activities and define the business and cost models for the customer acquisition. Additionally diverse service and support scenarios have to be worked out, prepared and initiated. This includes the preparation of diverse kinds of presentations and product training. Depending on the product setup and product, new or additional customer service and support facilities need to be planned and prepared. Apart from those product brochures, customer briefs and other collateral material needs to be designed and partl even printed before the technical product components become available.

Legal product components After deciding about the business, cost, support and service models the contractual framework for selling the new product can be created. These contract(s) need to cover e.g. the definition of the contract contents, IP disclosure plan, license scope, contracts for the services and payments, maintenance agreements, payment schemes etc. This framework needs the be prepared in parallel to the development to speed up the sales and roll-out process after the technical product components become available.

Deliverable	**Compulsory**	**Optional**
Process IP modules	X	
Performance Indicators	X	
Instantiation IP modules	X	
Design IP modules	X	
M&S product components	X	
Legal product components	X	
Stage and Project End report(filled template)	X	
Lessons Learned Report(filled template)	X	
Post-project review plan(filled template)	X	
Follow-up Action recommendations(filled template)	X	
Project Closure recommendation	X	
Req. for authorization to launch and roll-out	X	
Audit protocols		X

Table 3.19.: Stage 5 deliverables

3.4.15. Gate 6: Commercialization decision

The final gate determines the rate of success of the project and decides whether to commercialize the product. The gate validates whether the product is still commercial viable, whether the product components are of high quality and whether the customer acceptance is verified. It uses the Must-meet criteria catalogue presented in table 3.21. The central component is the checking of the completeness and ”positiveness” of all stage deliverables. The plans and deliverables of the M&S crew are reviewed carefully as well to launch the product in a controlled way. In the case of positive assessment, the gate converts the project results into a product and decommissions the project team. In cases of incomplete or to low quality deliverables, Stage 5 might be rerun to improve the shortcomings. A killing of the project in that stage might happen very seldom but could happen in cases of lost commercial or strategic viability.

Additionally the gate verifies the plans for post-project review and assesses the follow-up recommendations. The stage and project end report as well as the Lessons Learned report are reviewed. The approved Lessons Learned Report becomes publicized on the Intranet to allow future projects to learn from the experiences.

Table 3.20 shows the PRINCE2 process executed to perform the gate assessment together with the gates inputs and outputs.

Inputs	Activities	Outputs
Process IP modules Performance Indicators Instantiation IP modules Design IP modules M&S product components Legal product components Stage + Project end report Lessons Learned Report Post-project review plan Follow-up Action recommendations Project Closure recommendation Req. for authorization to launch and roll-out	DP5 Performing project closure	Approved product modules Approved Stage and Project End reports Approved and publicized Lessons Learned report Approved Follow-up plans Authorization to commercialize

Table 3.20.: Gate 6 activities with inputs and outputs

Criteria	met Yes	met No	N/A
Product still fits to the business strategy			
Market outlook(size, need, positioning) still suitable			
Business Case gives profitable outlook			
High quality Process IP modules			
High quality Performance Indicators			
High quality Instantiation IP modules			
High quality Design IP modules			
High quality M&S product components			
High quality Legal product components			

Table 3.21.: Gate 6 Must-Meet criteria

3.5. Process Development Methodology

In the targeted environment of MEMS process IP product development the methodology to develop the process IP is of major importance. To support and structure the process development efforts a new structured approach has been developed and is presented. The process development method is integrated as a subprocess into the overall development method, described in section 3.4. The method requires software support to provide for efficient and reproducible development. The software environment and concepts described in chapter 6 support these undertakings. In summary this solution provides support for the tasks of experimental verification integrated into an environment supporting the process Design For Manufacturability (DFM) and process simulation. The process DFM consists of the verification of the manufacturability of the fabrication sequence by process flow consistency checking. These checks allow to verify the process flow against abstract knowledge captured in rules which can be combined to model all kinds of restrictions e.g. disallowing higher temperatures in further processing, enforcing certain pre- or post-processing of a step and so forth. After a satisfying assessment via the consistency checks, a process simulation marks the next, more detailed verification step. The simulations can be performed on several detail levels (e.g. geometric, empirical, physical) to allow for appropriate granularity. Combining this with the support for experimental verification a round trip integrated solution is created easing, supporting and capturing the MEMS process developments. This environment is described in more detail in chapter 6 and in the publications [WPS+05], [WSP+06] and [SOH+06]. It is suited to support the proposed process IP product development methodology described in following sections.

3.5.1. Demonstrator approach

Analyzing different fabrication lines it can be concluded that each line uses different equipment or slightly different calibration of the same equipment. For microelectronic fabrication these deviations are less of an issue but for the manufacturing of silicon based MEMS these differences have a high impact. This is due to the additional mechanical requirements. Therefore it can be concluded that an ”of-the-shelf process IP product” is hard to develop. This is due to the potentially large property changes of the ”same” material after porting and integrating a process IP module at a customer site. It is more realistic to use a ”Product demonstrator” approach, because the process (sub-)flow integration and implementation (materials, machines, etc.) is unknown. This approach is described in more detail below. An ”of-the-shelf product” might only be achieved by maturing an IP product over time after having ported it to several similar fabrication lines.

The ”Product demonstrator” approach is based on the following ideas:

- Develop a ”Product demonstrator” showing that is it doable on silicon in an industry line or an industry based institute and generate the knowledge how to use this demonstrator.

- Evaluate results based on the ”Product demonstrator” for the areas of device and chip characterization, determination of process window, device and chip yield and device and chip reliability data.

- Build up knowledge around the ”Product demonstrator” (cause-effect chains, impacts of parameters, What-if, etc.) which should help to speed up the re-development and knowledge transfer during the convey of the IP product to a customer line.

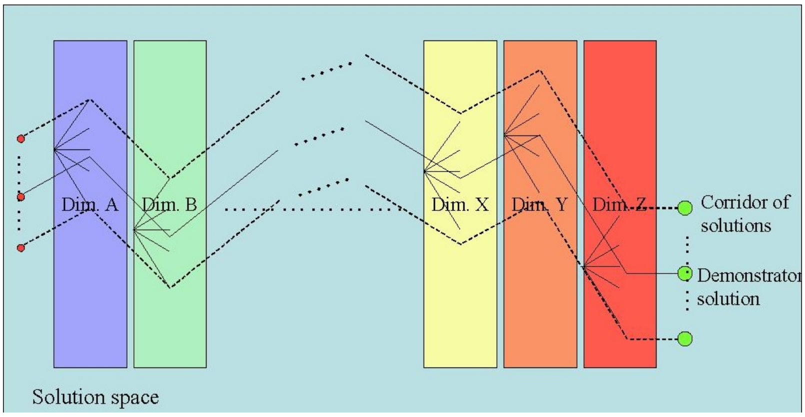


Figure 3.16.: Process Development Methodology

Within the demonstrator approach the achieved solution can be considered a single point or line solution where only exactly this recipe in this environment works. This means that just this solution can be shown to customers and proof the product and device idea. Variations of the process recipe are not yet tested and assessed or only to a very small degree. To make the solution more robust, the process window for each process step needs to be widened to allow for variations at a location the process is ported to. The characterization for yield, reliability, lifetime, endurance etc. of devices manufactured with the single line process can be performed in parallel with the widening efforts. Within the product development method described in section 3.4, the demonstrator prototype (single line solution) is developed within Stage 4 and the characterization and further developments are performed in Stage 5.

3.5.2. Two development phases

The overall process development is performed in two phases. The first phase develops the above described single line solution exploring only a relatively small set of possible combinations of degrees of freedom. Figure 3.16 presents the degrees of freedom or dimensions as coloured boxes. For each of these dimension a small set of discrete points and their combinations are build and tested during the first phase of the development. These discrete points are shown as the fan-outs in the coloured boxes in the figure. This first phase develops in the depth, finding a single point or very narrow set of solution points for every dimension of the solution space (geometrical (width, length, thicknesses), process (flow rates, materials, etc.), etc.). The approach implies that in this first phase only a small set of paths will be developed until a working ”Product demonstrator” is achieved.

The knowledge around the single demonstrator solution or small set is build during an extension phase. This second phase widens the multiple paths of the initial solution into a corridor by exploring interesting dimensions further and comparing the results against the already achieved working demonstrator. For that, each dimension of the solution space (degree of freedom), e.g. single material, is explored into the width to build up a bandwidth or process window for each of the possible and sense full dimensions. This adds to the knowledge base captured by the above mentioned process design and tracking system. The wider process windows enable better portability and more potential customers. The achieved discrete points of each dimension of the solution space are considered as a bandwidth (process window) with minimum and maximum boundaries. Potentially there might be some bandgaps implying that simple interpolation for the solution points might not be feasible in all cases.

3.5.3. Build a process window

To perform the width exploration of each feasible dimension in the second phase of the development, it is necessary to know the dependencies between the different possible settings for a single step. This requires to have inspected the cause-and-effect dependencies between the different degrees of freedom. To evaluate the cause-and-effects dependencies a Design of Experiment (DoE) approach can be used. DoEs, summarised in [Sta05b] and described in more details in [ReV04], extract the major dominating parameters concerning a set of effects. Design of Experiment takes care of reproducibility and unnecessary repetition issues as well as coverage issues. Additionally some DoE methods provide for strategic experiment reduction (fractional factorial design) to extract the required dependencies with less experiments than full factorial approaches. These techniques trade in knowledge gain and full information retrieval for efficiency gains. Using these techniques, significant savings in time and effort can be gained because of the reduced number of necessary experiments to extract the main dependencies. To be able to use this kind of approaches it is essential to plan the efforts using appropriate tools like MiniTab [Min05] to be able to cautiously and strategically plan the experiments.

Having designed a set of experiments, the amount of physical experimentation can be reduced even further. To limit the overall amount of experimental verification, the system described in chapter 6 allows for a three step verification approach, first checking the principle manufacturability, then performing process simulations and afterwards supporting the experimental verification. A precondition for using this approach is to put the focus on conserving and making accessible the acquired knowledge to become traceable and reproducible and have full control over the knowledge base. This is supported by the process development development and tracking system introduced in chapter 6.

The extracted dependencies and gained knowledge can be used to adept the process parameters and to widen the process window getting a more robust process step or dimension. Combining all successful experiments a corridor of knowledge can be build. This scenario is sketched in Figure 3.16 where a corridor of possible solutions is build around the single line prototype solution.

3.6. Conclusions

The newly introduced product development methodology combines two standard methods into a specifically tailored method. Together with the formalized process development method it is especially suited for MEMS process IP product development. The newly developed and proposed method supports the introduced business model in an optimal way and enables to drive an IP product idea to a marketable IP product. Therefore it addresses directly the requirements for a PE method, summarised in section 2.3.2. To be able to develop a product using this method an environment supporting and enabling this method needs to be established. Such a Product Engineering Framework is introduced in chapter 4.

4. Product Engineering Framework

The development methodology described in section 3 requires significant tool support for its processes to be efficient. Additionally these tools need to provide certain features that are required for a Product Engineering Framework targeted to successfully support IP product development. This chapter describes the necessary features extracted during studying such a business environment. Additionally it provides a view from the business perspective, identifying the areas of concerns which need to be covered by the different tools of such a framework. Taking the business perspective, the different building blocks (a.k.a. bricks) are described step by step and tools supporting these areas of concern are introduced thoroughly. An overview of the areas of concern is depicted in Figure 4.1.

For the further descriptions it should be noted, that the possible tools typically address several areas of concern so that a sharp distinction is often difficult to achieve. This is due to the overlap of the different areas. Additionally it is required to look at an aspect from several perspectives. On the other hand this overlapping provides a pre-integrated environment so that less components need to be integrated to form an overall framework.

4.1. Required PEF Features

The analysis of fabless IP companies developing process and design IP revealed several required features of a Product Engineering approach and its framework. These features can be summarised into:

- Enable knowledge management, especially for the core IP parts by management of the knowledge acquisition and management of usage and access.
- Enable reproducibility of design and process development efforts by complete version control covering all necessary areas, input/output traceability and complete data acquisition.
- Empower efficiency by easy access, early problem detection, regular status and so forth with a slim but sufficient, extendable and scalable approach, a structured and well defined methodology and limited overhead.

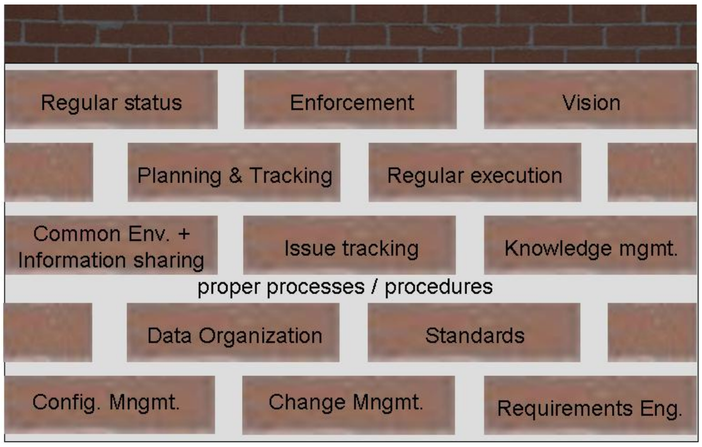


Figure 4.1.: Necessary bricks for a Product Engineering Framework

- Give or increase control about status, progress and issues by the usage of planning & tracking, constant status update and issue awareness.
- Ensure to be target focused by sharing and communicating a vision, clear definitions, targets and responsibilities and active issue tackling.
- Enable easy and efficient communication by shared access to information and knowledge and synchronous and asynchronous communication means.
- Aid better overall coordination by project management and clear responsibilities.

This poses the the major question "How to best achieve all this?" which can be briefly summarised by:

- Use structured, proven and well defined approach
- Always execute along the same procedures
- Do **not** cut corners
- Constantly be aware of target vision, current planning, current status, known and expected issues, responsibilities etc..

The defined approach, the execution along the same procedures and the prevention of corner cuttings are covered by the processes and procedures and integrated quality controls of the product development methodology described in section 3.4. The later points introduce areas of concern, summarised in figure 4.1. These aspects need to be taken care of by the Product Engineering Framework components. The areas are carefully explored in the following sections looking from the business perspective and introducing possible tool components for each area of concern.

4.2. Configuration Management

4.2.1. Features provided/addressed

Configuration Management (CM) is one of the cornerstones of a Product Engineering Framework because it is one of its baseline components addressing

- reproducibility, by adding version control
- control, by storing the changes and the authors of the changes and
- communication/coordination, by building a communication platform.

Additionally Configuration Management is a central component of the development methodology as outlined in chapter 3.

4.2.2. Definitions

Several slightly different definitions of the terms Configuration Management can be found in literature. The definition herein is based on the quite generic definitions of Configuration Management in [Wik05d]. Further aspects have been added from literature e.g. from [Ber01]. Analyzing these definitions and following the definitions in the IT Infrastructure Library (ITIL)[1], it can be noted that Change Management, described in section 4.3, and Configuration Management are closely related processes.

Although Change Management and Configuration Management are closely related, here they are described as separate building blocks. This is because both bricks are closely related to requirements engineering as well. Change Management is described in 4.3 and Requirements Engineering is described in 4.4.

While the definitions in the above citations are relative generic and abstract, the term Configuration Management is defined in more detail for the Software development area. There are several aspects to CM which are not explicitly covered by these generic definitions. According to [Ber01] Configuration Management for software development needs cover the aspects of Configuration Identification, Configuration Control, Status Accounting Audit and Review. Additionally it needs to contain Build Management, Process Management and Team Work. These aspects are explained in more detail in section B.1.

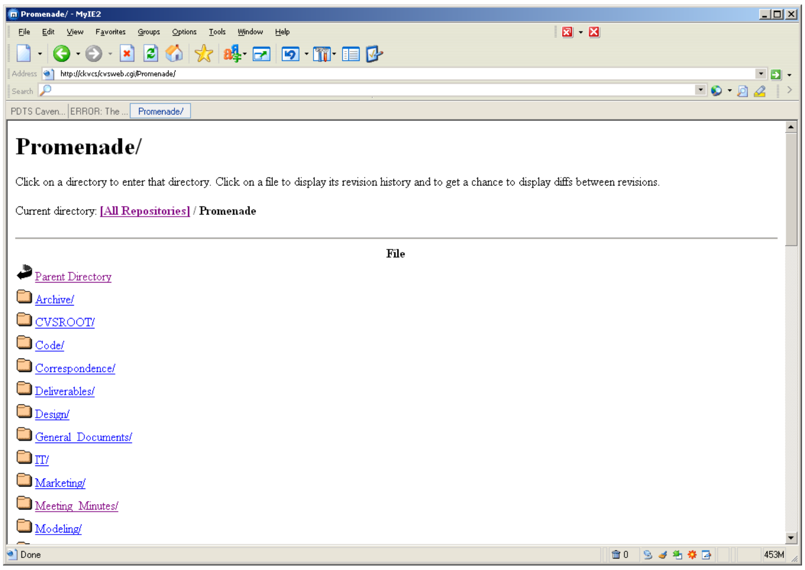
[1] The ITIL is coordinated and published by the Office of Government Commerce of the United Kingdom (OGC), [Off05], and defines business process and components for IT Service Management. It can be considered as a reference for best practices in the area of IT support. A good and brief introduction into the ITIL can be found in [Wik05f].

Although these definitions are targeted for software development and source code management, the mentioned issues are generic and should be applied to all kinds of development efforts. It is secondary that for software developments the source code builds the deliverable and product itself. This is not necessarily the case for other product development areas. For the herein considered IP development all issues raised above are 100% valid. They apply to the documentations, test results, and so forth of an IP product as well. While the first block of the above aspects is considered to be the core of CM, the issues of the second aspect block will be attributed to other building blocks/bricks. Altogether these points build a central focal point for the overall definition of the Product Engineering Framework.

4.2.3. Components for Configuration Management

Component:Revision control

The aspects of Configuration Identification, Configuration Control, Status Accounting Audit and Review are all covered by standard tools for Revision Control. Plenty of software tools address the issue of Revision Control e.g. IBM's (former Rational) ClearCase [IBM05c], GNU RCS [Fre05d], GNU CVS [Fre05b]. For a slim, fabless environment heavyweight systems like ClearCase can be considered overkill. Because the system should operate server based and, - if possible - be free of charge. It is defined to use the Open Source system CVS for Revision Control.


Promenade/ - MyIE2
File Edit View Favorites Groups Options Tools Window Help
Address http://ckvcs/cvsweb.cgi/Promenade/
Search
PDTS Caven... ERROR: The ... Promenade/

Promenade/

Click on a directory to enter that directory. Click on a file to display its revision history and to get a chance to display diffs between revisions.

Current directory: [All Repositories] / Promenade

File

Parent Directory
Archive/
CVSROOT/
Code/
Correspondence/
Deliverables/
Design/
General_Documents/
IT/
Marketing/
Meeting_Minutes/
Modeling/

Figure 4.2.: Repository browser CVSweb

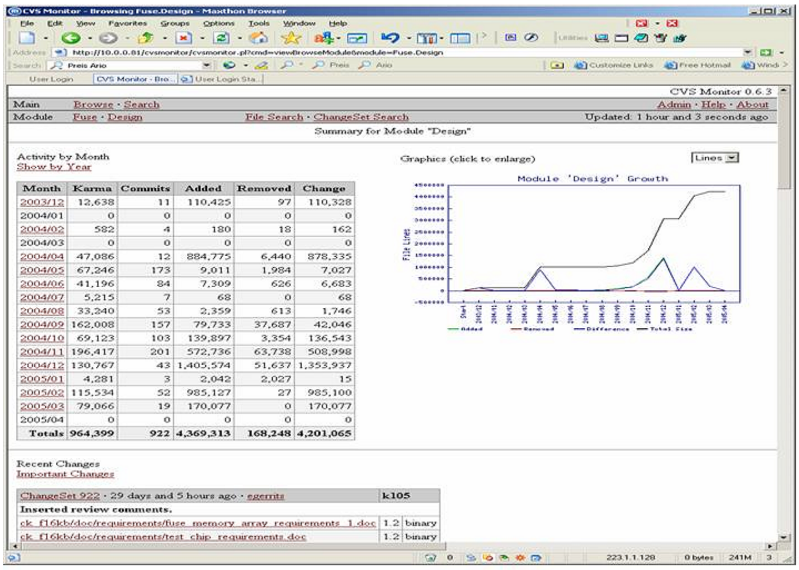
Component: Repository browsing

As a useful addition to the basic CVS system the tool CVSweb [Fre05e] has been integrated into the framework. This tool addresses team work and easy collaboration. CVSweb enables browsing, like presented in figure 4.2, and URL linking the CVS repositories and modules. Especially the URL linking and direct downloading via the web interface is useful because it provides version controlled and save referencing. This prevents dangling links in publicized Intranet or E-Mail contents. Additionally CVSweb provides functionalities to easily present the version graphs, allowing Configuration Identification, as well as difference tools to show the changes between versions of text files.

Component: Repository change monitoring

As indicated above and described in [Ber01] on important area for Configuration Management is Status Accounting Audit. This can be achieved by change monitoring of the repository and change presentation to

- be steadily informed about changes done to the different projects
- operate as an entry point for questions like: 'Which items have been changed when?'
- present the changes over the timescale on different levels (repository, module).



Updated: 1 hour and 3 seconds ago

Summary for Module "Design"

Activity by Month

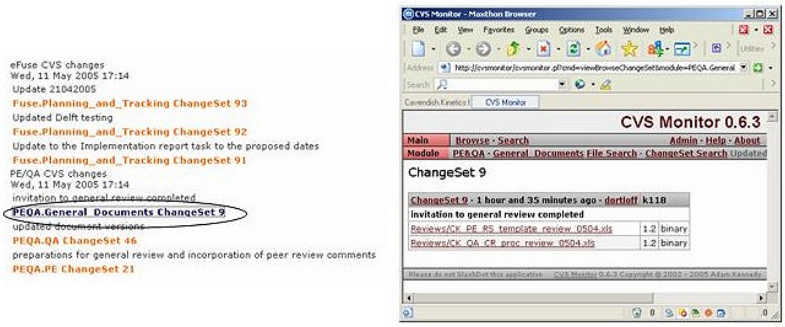
Month	Karma	Commits	Added	Removed	Change
2003/12	12,638	11	110,425	97	110,328
2004/01	0	0	0	0	0
2004/02	582	4	180	18	162
2004/03	0	0	0	0	0
2004/04	47,086	12	884,775	6,440	878,335
2004/05	67,246	173	9,011	1,984	7,027
2004/06	41,196	84	7,309	626	6,683
2004/07	5,215	7	68	0	68
2004/08	33,240	53	2,359	613	1,746
2004/09	162,008	157	79,733	37,687	42,046
2004/10	69,123	103	139,897	3,354	136,543
2004/11	196,417	201	572,736	63,738	508,998
2004/12	130,767	43	1,405,574	51,637	1,353,937
2005/01	4,281	3	2,042	2,027	15
2005/02	115,534	52	985,127	27	985,100
2005/03	79,066	19	170,077	0	170,077
2005/04	0	0	0	0	0
Totals	**964,399**	**922**	**4,369,313**	**168,248**	**4,201,065**

Recent Changes

ChangeSet 922 · 29 days and 5 hours ago · egerrits	**k105**	
Inserted review comments.		
ck_f16kb/doc/requirements/fuse_memory_array_requirements_1.doc	1.2	binary
ck_f16kb/doc/requirements/test_chip_requirements.doc	1.2	binary

Figure 4.3.: Repository monitoring with CVSmonitor

A tool to achieve this is CVSmonitor [Ali05]. CVSmonitor regularly monitors the activities on different repositories and modules, and presents the changes graphically and in change sets. An example is presented in figure 4.3. The graphs present the change rates over time and make the activities visible from different perspectives. Additionally it provides features for browsing the changes, searching of files, changes and dates. Another very nice feature is that it generates RSS feeds presenting the change sets. These can easily be integrated into an Intranet presentation showing the last changes ordered by time. Because these change sets monitor on one module only, a useful addition of a RSS combiner script has been implemented. This script aggregates the changes per module to changes per repository. This provides an overview per project which can easily be presented on the Intranet. Figure 4.4 presents such an combined RSS feed on the left side of the figure. From these feeds it is possible to directly link into the change sets to see the details of the change. Following the link of the marked change on the left hand side of figure 4.4 guides to the right hand side of the figure showing the related change set. Having the changes visible on the Intranet allows to easily view them which improves team communication and control.



eFuse CVS changes
Wed, 11 May 2005 17:14
Update 21042005
Fuse.Planning_and_Tracking ChangeSet 93
Updated Delft testing
Fuse.Planning_and_Tracking ChangeSet 92
Update to the Implementation report task to the proposed dates
Fuse.Planning_and_Tracking ChangeSet 91
PE/QA CVS changes
Wed, 11 May 2005 17:14
invitation to general review completed
PEQA.General_Documents ChangeSet 9
updated document versions
PEQA.QA ChangeSet 46
preparations for general review and incorporation of peer review comments
PEQA.PE ChangeSet 21

CVS Monitor - Maxthon Browser

http://cvsmonitor/cvsmonitor.pl?cmd=viewBrowseChangeSet&module=PEQA.General

CVS Monitor 0.6.3

Main | Browse · Search | Admin · Help · About
Module | PE&QA · General_Documents File Search · ChangeSet Search Updated

ChangeSet 9

ChangeSet 9 · 1 hour and 35 minutes ago · dortloff	k118	
invitation to general review completed		
Reviews/CK_PE_RS_template_review_0504.xls	1.2	binary
Reviews/CK_QA_CR_proc_review_0504.xls	1.2	binary

Please do not SlashDot this application   CVS Monitor 0.6.3 Copyright © 2002 - 2005 Adam Kennedy

Figure 4.4.: Combined RSS feeds for a repository with marked link

4.3. Change Management

4.3.1. Features provided/addressed

Change Management is another fundamental building block of a Product Engineering Framework because it is addressing

- reproducibility, by organizing change and the application of changes
- control, by assessing changes and
- communication/coordination, by communicating approval of changes.

Additionally Change Management and Change Control is a central component and technique of the development methodology as outlined in chapter 3 and depicted in figure 3.6.

4.3.2. Definitions

[Wik05b] gives a quite generic definition of Change Management. Analyzing this definition and following the definition in the IT Infrastructure Library (ITIL; [Off05]), it can be noted that Change Management and Configuration Management, described in section 4.2, are closely related processes. Additionally those two are closely related to requirements engineering described in section 4.4.

Change Management, defined in e.g. [Wik05f] and [Wik05c], and Configuration Management, e.g. defined in [Wik05f] and [Wik05d], can be distinguished in the following way. Change management focusses more on the overall coordination of changes while Configuration Management focuses more on the tracking and individual changed configurations. Therefore the Change Management takes care of the big picture and uses the Configuration Management to control the changes.

The aspects of Configuration Management are described in section 4.2 while the minimal necessary components of the Product Engineering Framework for Change Management are described below.

4.3.3. Components for Change Management

As introduced in the Definitions, Change Management focuses more on the general coordination of changes. These can be changes in the infrastructure, as described in Change Management process of the ITIL [Wik05f] and [Wik05c], or changes applied to a project or product. Therefore the change management needs to foresee mechanism for the whole life cycle of a product, managing changes applied during the developments or the maintenance of a product. It needs to contain tool and process components. A minimal set is described in the following paragraphs.

Component: Change Request (CR)

To modify an already existing product or a product development project it is essential to provide a predefined Change Request template. Such a template needs to cover e.g. the Object to change, the general reason(s) for the change(s), a detailed description of the change(s), prioritisation of the change(s). All necessary aspects to cover are described in section A.3.6. The template and the change requests produced using this template need to be included in the projects repository.

Component: Change Request procedure

The purpose of the change request procedure is to give the engineering and quality framework predefined, easy to execute means to achieve controlled change of already internally or externally released deliverables or to products currently under development. Because of that it builds one component of the Change Management and creates the gateway to propose, evaluate and execute requested changes. The Change Management together with the Configuration Management takes care of keeping the CRs and the different configurations as well as the requirements for a project synchronized. The change request procedure provides a framework to efficiently manage the change requests and apply them to the targeted deliverable. The CR procedure needs to tie in with other components of the Quality Assurance Framework described in more detail in section 5.

Component: Change Monitoring

Change monitoring belongs to Change Management as well as to Configuration Management. Because it needs to be integrated with other components of the Configuration Management it is described in the context of Configuration Management and can be found in section 4.2.3. Tools for this purpose are described there as well.

4.4. Requirements Engineering

4.4.1. Features provided/addressed

Requirements Engineering (RE) and resulting requirement specifications are the initiator of a product vision, are used for calibrating a product idea against the market needs or are analyzing a market need to generate a product vision. It is one of the baseline components addressing

- target focus, by defining the goal,
- communication/coordination, by communicating the aims and the constraints and
- control, by early definition of the goal and its constraints together with defining an orientation point during the developments.

Requirements Engineering uses the requirements specification to track the requirements through the development cycle.

4.4.2. Definitions

Requirements define what features a product should provide to the real world and define the products constraints. As defined in [NE00] and [Zav97] they can be considered as the part concerned with anchoring development activities to a real-world problem. If requirements are not defined or not properly defined this typically leads to late delivery, false assumptions, inappropriate products and so on. Additionally problems in the requirements definition cause change requests after a product is already under development, in roll-out or already in use.

An overview introduction to requirement engineering and its techniques is given in [Wik05k] while [NE00] and [GK98] go into much more detail. Additionally the later two give much more insight into the social aspects, the sequences, procedures and different aspects of RE. The paper [NE00] and the book [GK98] have been summarised in [San05] and [Ger05] respectively.

As a summary it can be said that Requirements engineering is concerned with eliciting, analyzing, and documenting the product requirements taking the whole product into consideration. This includes all types of deliverables belonging to a product e.g. documentation, plans, physical products and so forth. To document all requirements and constraints (non-functional requirements) for a product, a requirements specification is put together by marketing and sales personal. The generation of this specification is coordinated by the requirements engineering role. This document builds the baseline product documentation and sets the expectations on all sides, the customers and marketing, the engineers and management. It includes a product overview, a glossary, the statement of the functional requirements and the operational constraints. A more detailed description of such an documentation structure in given is the component Requirements Specification below. The requirement specification is closely related to the PE building block ”Vision”, described in more detail in section 4.14, because it enables the creation of the product vision.

To support the requirement engineering efforts tool support is advisable. The most essential part of such tools is a template for the requirement specification.

4.4.3. Components for Requirements Engineering

Component: Requirement Specification (RS)

The one component laying the baseline of a product development and its different versions is the requirement specification. A template defines the different angles and perspectives required to be evaluated to guide and support the requirements elicitation. An overview about the defined template is given in section A.3.3. This baseline laying specification needs to be stored in the folder

”Specifications” of the standard electronic binder (described in 4.5) and has to be under revision control (4.2) to enable reproducibility and communication.

Component: Requirements management tools

Requirements management tools support the requirements engineers in their diverse activities. They offer functionalities to more easily elicit requirements, coordinate the requirements elicitation process, consolidate requirements between several documents and enable traceability between requirements and deliverables. Because most of the tools spin off from software engineering frameworks, a careful selection needs to be made to find an appropriate solution. A comprehensive list of requirements management tools is published in [Lud05] from which IBMs Rational RequisitePro [IBM05d], Telelogics Doors [Tel05] and Sparx Systems Enterprise Architect [Spa05] are the most prominent. As mentioned in [Lud05] these tools are very useful but highly depend on the quality of the requirements. An even earlier precondition for successful requirements engineering is the awareness of the importance of requirements engineering on all levels of the organization.

4.5. Data Organization

4.5.1. Features provided/addressed

The usage of a centrally defined data organization scheme (e.g. electronic project binder, central revision control system) which is shared between all projects and all engineers adds to the features of

- Reproducibility, because all ’once published’ information can be easily retrieved again from standard structures,
- Efficiency, because all engineers know where to search for certain information and
- Easy and efficient communication, because everybody is aware of the standard place where certain information is located and can retrieve from it easily.

Because of that the usage of a centrally defined data organization is one important building block of the Product Engineering Framework.

4.5.2. Definition

As outlined in the features section, the data organization brick needs to contain methods, tools and conventions on how the project development and project organizational data is stored. The central idea is to organize the data of all types of projects (internal projects, customer projects, etc.) in always the same structures. Therefore the storage scheme needs to be able to handle a 3-tier scenario where the organization in question is in the middle and it has customers on the one hand and is a customer to other companies on the other hand. This scenario adds to the structuring and categorization of all kinds of data and to the above described features. Additionally this structure should be integrated, loosely or directly, with the Configuration Management to be able to keep track of all versions exchanged with customers. There should be one central place where all data is stored for central reference of all engineers. The engineers work spaces contain only a subset of the central space (sandbox of the CM). At least all project deliverables need to be registered into the CM while the engineers sandbox might contain additional temporary files. Via tagging of releases in the Revision Control System it is possible to manage different subsets of data. This enables that even intermediate or helping files can be checked into the version management but will not be contained in official project releases because of their tags. The communication between the different workspaces and the central project binder should be done via the CM which should be executed on a regular basis.

Some general recommendations about the electronic project binders are that the generated structure is mandatory but can be extended and structured further down, that sensible filling is the responsibility of the project manager or the group managers, that every customer and supplier should have separate subdirectories (3-tier scenario) and that the overall contents of the electronic project binder must be legally complete → it might be necessary to scan documents e.g. contracts.

4.5.3. Components

Component: Electronic project binder

The integration and similarity assurance is achieved by a set of scripts realized in the GNU bash [Fre05a] scripting language. The scripts generate a standard electronic project binder on the shared project servers accessible from all platforms and systems. Additionally these scripts generate a new repository with modules in the Revision Control system. The electronic project binder needs to contain sub folders for Correspondence (in/out), Deliverable(in/out), General Documents, Meeting Minutes(external/internal), Offers/Quotes (external/internal), Project Management(all project plans, project controlling information (budget, plans, etc.)), Specifications (in/out) and Work areas for the different business groups. When using the project creation scripts, the binder and the repository created will automatically contain all predefined project management templates.

Component/Brick: Central Configuration Management

For organizing the data, it is required to the set up a central Configuration Management component as described in section 4.2. As described in that section, this should be done via one central Configuration Management server responsible for the Revision Control allowing repository browsing and monitoring. To integrate both data organization means it is advisable to set up one repository per project/product and make all folders of the electronic project binders a separate module of the repository. This way it is possible to limit the sizes of the sandboxes and to have all necessary parts of the project binders under control of the Revision management.

4.6. Standards

4.6.1. Features provided/addressed

The usage of existing standards is another central point for a Product Engineering Framework and can prevent to reinvent the wheel and establishes the usage of best practices. The usage of existing standards can provide the following features:

- Easier communication and sharing, by talking about common things,
- Efficiency, because of using already proven approaches and preventing reinventing the wheel

Additionally the usage of best practices is fostered, the team efforts are streamlined and structured and earlier and easier "buy-in" is enabled.

4.6.2. Definition

In general the usage of as many Standards as possible is preferable because of credibility and efficiency reasons. The area of Standards can be divided into the publicly defined and accepted standards and internally defined standards. While publicly available standards are always preferable, internally defined standards are necessary additions to fill not covered areas.

4.6.3. Standard Components

There a plenty of different publicly available standards which are applicable and can be used for a Product Engineering Framework. Only a few examples for the different application areas are listed below because the selection is very company specific:

- Building blocks for the Engineering Framework e.g.

  - Stage-Gate™ – product development method used for the product development methodology described in chapter 3.4 and introduced by [Coo98].
  - PRINCE2 – project management method used for the product development methodology described in chapter 3.4 and documented in [GC02].
  - ITIL – standard to provide IT and other services. The ITIL can be used after the needs to support customers arises, providing processes for customer support and is described in [Off05].
  - CMMI – standard to define, capture and improve the maturity of an organization and its development processes, described in [Car05].
  - Standardized tools and environments like MS Exchange [Mic05e], CVS [Fre05b], CVSmonitor [Ali05], CVSweb [Fre05e], Bugzilla [Bug05], Mambo [Ope05a] and so forth.

- Process Development Standards e.g.

  - Materials standards
  - Packaging standards
  - Test standards

  many of them published by SEMI [Sem05].

- Design Standards e.g.

  - Standardized design flows
    analog: e.g. Specification, Schemantic Simulation, Layout;
    digital: e.g. Specification, HDL behavioral model; HDL RTL model; RTL Sim., gate level synthesis, place&route
  - Standard HDL like VHDL + Verilog
  - Verification techniques like: Design Rule Check(DRC), Layout vs. Schematic(LVS), Parasitic Extraction (PEX), Static timing Analysis (STA), Simulations on different levels (e.g. Spice, SystemC, VHDL)
  - Corner Simulations
  - Test standards like JTAG boundary scan path, IDDQ test, Reliability analysis (e.g. temp. cycling, etc.)
  - EDA tool exchange formats e.g.
    GDSII [Buc05], OpenAccess [Sil05a]

- Software Coding standards e.g.

  - JAVA coding guidelines/conventions [SUN05a]
  - JAVA GUI development guideline [SUN05b]

  - C++ Ellemtel rules [Ell05]

Examples for necessary internal standards are:

- Templates for diverse purposes e.g. product deliverables in different formats (Word, Excel,...), requirement specifications, change requests, etc.
- Checklist for e.g. tape-outs, product-assembly, etc.
- Procedures for e.g. deliverable reviews, change request handling, signing-off a product, etc.

Altogether the usage of standards makes the Product Engineering Framework easier to introduce, more efficient and more reliable.

4.7. Common Environment + Information Sharing

4.7.1. Features provided/addressed

The usage of a Common Environment shared between all engineers and a platform to share information adds to the features of

- Reproducibility, because all once published information can be easily retrieved again,
- Efficiency, because all engineers are on the same page about, status, news etc.,
- Easy and efficient communication, because everybody is retrieving his information from the same sources and
- Target focused, because the key deliverables are constantly visible.

Because of that the usage of a Common and Information sharing Environment is one important building block of the Product Engineering Framework. Here two different approaches can chosen. The best of breed approach choosing the best tool for each area versus the all tools from one vendor approach. Both approaches have pros and cons. The best of breed approach has the advantage to get the best tools for the different areas but often causes difficulties when integrating the different tools. If choosing all tools from one vendor the tools are typically well integrated but not all tools might be leading edge.

4.7.2. Definition

Overall it is important to provide a common environment for all users to ease the communication and usage and to enable sharing of information, control and reproducibility. This needs to be true especially for distributed sites to ease the data exchange, inter-site travel and communication. While systematic capture of knowledge and information, described in section 4.9, is only one side of the medal, the sharing, searching and easy retrievability is the other side. One important component is to provide on central, generally used communication platform able to integrate all other framework components. As an integrating, publishing and sharing platform Intranet based solutions are state-of-the-art. Because of that all tools of the framework are accessible via the Intranet or are even integrated into it. This approach has been previously reported e.g. in [DBG+03], and has been adopted in a similar manner.

Such a platform can be used to easily publish all kinds of information from news over status and so forth. An additional facet for the sharing and easy accessing is the common organization of project related data. The data organization, described in 4.5, adds to the sharing of information as well because it places data in well known structures so every engineer can easily retrieve it. Another necessary infrastructure component are the facilities to easily communicate, especially for multi-site organizations. Such facilities need to include an overall groupware solution, conference call facilities and means to work between sites via online collaboration. These facilities are important to share information across the organization.

The aspect of a Common Environment has even more importance for the contributions of the core IP modules of design and process knowledge. The process development needs to be supported in a special way to capture the knowledge and make the efforts efficient. This can be achieved with a novel design, simulation and development tracking tool which is described in detail in chapter 6 and which has been integrated into the product engineering framework. The environment setup for the design IP requires quite some groundwork as well. Key is a shared setup of the complete EDA environment as described in e.g. [DBG+03].

4.7.3. Components

Component: Intranet

As described above, establishing an Intranet as an integrating information exchange platform is important. Because every engineer should be able to publish his news and data on this platform, an easy to manage base system for the Intranet creation and management needs to be used. For this purpose Content Management Systems (CMS) are state-of-the-art because they allow even users unexperienced with Internet publishing technologies to easily create and publish their own contents. There are plenty of commercial and Open Source Content Management System available on the market e.g. Microsoft Content Management Server [Mic05b], Mambo Open Source CMS [Ope05a] and so forth. For a slim and cheap framework Mambo Open Source is a good choice because it is easy to manage, easy to use and only very limited training for publishing own content is necessary. It is based on standard PHP technology and is free of charge. Additionally it supports pluggable components and modules making it very flexible and extendable.

Component: Basic environment

Besides the integrating Intranet platform the following basic environment tools and components are necessary to provide the basis for a Common Environment.

- Potentially mixed environment of different operating systems, like Linux and Windows, to provide the best platform for specific needs. For remote access to machines VNC or TightVNC [Sou05], PuTTY [Ope05b] or MS Remote Desktop [Mic05d] can be used.

- Shared, generally accessible file servers having the possibilities to establish detailed access control on specified areas. This is required to protect confidential customer specific data from general access and makes it accessible to project members only.

- Groupware solution for E-Mail, calendar, task, ... sharing like Microsoft Exchange [Mic05e], IBM Lotus Notes [IBM05b] or the Open Source package eGroupWare [eGr05]

- Generally accessible and integrated revision control systems, like described in 4.2

- Easy work space movement by enforcing server based work style and roaming or clustered profiles

- Generally applied login scripts, ensuring consistent baseline user setup under all environments e.g. Windows and Linux

- Means to facilitate easy multi-site communication and work by conference calls, video-conferences and collaborative tool usage e.g. MS Netmeeting [Mic05c], WebEX [Web05a]

Newer developments are targeted to integrate some of the above aspects into one solution addressing several needs. One example for such approaches is the Oracle Collaboration Suite [Ora05a] with its diverse components. While such a solution is very handy, for a company with limited resources such a tool-suite is far to heavy and expensive.

Component: Process development setup

Additional support tools for the generation of the core IP modules are necessary. As defined above and described in chapter 6 the setup of the process development environment requires special care because it generates one of the major blocks of MEMS IP. Because this setup is described in more detail in chapter 6, only the environment for the generation of the design IP is described in more details in this section.

Component: Design group setup

As described in [DBG+03], a framework for a common design environment needs to be set up. This needs to initialize the users environment in a unified and centrally for the project organized manner. This includes the selection and definition of technology files, support tools and their specific versions and project specific repositories to perform revision control. All resources should be shared and the paradigm ’single master source of information’ (e.g. CVS repository) needs to be constantly applied. Otherwise the management of the different sources becomes too complex causing confusion and potentially incomplete deliverable and knowledge collection.

As a design tool environment for silicon based MEMS a similar approach and tools selection like for VLSI design activities can be assumed. This implies that typically a mixture of EDA tools from different vendors like Mentor Graphics [MG05], Cadence [Sys05b], Synopsis [Inc05] and Silvaco [SIL05b] exists. Because all these tools require special setups and need to coexist next too each other, the special efforts described in section B.2 need to be considered. The environment described there uses a script based, centrally managed project initialization environment setting the work space of all engineers.

The philosophy behind the approach is that the projects ’tool smith’ has full control about the settings used for the project or project version. By adding a new initialization script a different setup can be used. This is of special importance for the lifetime of the project and product because initial project setups can be maintained while a newer project version might use different versions of tools, different technologies, additional tools and so forth. With the rule that EDA tools once used for a project release will never be uninstalled, only archived, it is possible to become version save and reproducible, especially by checking the tools into the revision control system as well.

4.8. Issue Tracking

4.8.1. Features provided/addressed

Issue tracking provides for awareness of problems together with their criticality. It builds a central vehicle to manage and resolve issues. Therefore issue tracking addresses the following PEF features:

- Control, by being aware about the open issues and their criticality,
- Target focused, by knowing which items to tackle and
- Knowledge Management, by storing resolutions for the issues.

It is a central part of the exception treatment of the development methodology as well.

4.8.2. Definition

The term issue in the current context refers to several topics. Normal questions or comments can be considered issues to be resolved because they require attention and can have major resource impacts. Questions can be posed by customers, which typically have a higher priority, or can be raised by internal personal. Other kinds of issues are problems during the development process posing additional project risk, defects, bugs, incidents of product deliverables and so forth. In summary everything unforseen requiring attention and a solution can be considered and issue.

In [Wik05e] the issue definition describes the following additional attributes of an issue: *Each issue may have an urgency value assigned to it, based on the overall importance of that issue. Critical issues are the most severe that should be resolved in the most expedient way possible, taking precedence over all other issues. Low or zero urgency issues are minor, and should be resolved as time permits. Other details of issues include the customer experiencing the issue (whether external or internal), date of submission, detailed descriptions of the problem being experienced, attempted solutions or work-arounds, and other relevant information.*

4.8.3. Components

For the Product Engineering framework two areas of tool components can be foreseen, Issue Tracking Systems and the FMEA method/tools.

Component: Issue Tracking Systems

An Issue Tracking System is designed to manage issues having the properties summarised in the definition above. Again [Wik05e] gives a good, generic definition of the characteristics of an issues tracking system which should not be repeated here. In summary these tools enable the communication about issues, provide for assigning the issue for resolution to a certain resource and keep track of all changes, discussions, etc. to such an issue item. Especially they typically capture the resolution for an issue and therefore are a building block for the knowledge management (refer to 4.9) as well.

There are plenty of different issue tracking system available on the marketplace, commercial as well as Open Source. They all have their pros and cons and to find the right solution highly depends on the specific company needs and its other existing tools. This is especially true because the gradients between issue tracking systems, trouble ticket systems, support call tracking systems are smooth. Tools of these areas all cover partly the other areas as well and therefore the choice needs to carefully take the specific needs into account. The costs are another aspect for this decision because the prices vary and free of charge Open Source systems exist as well. Examples for commercial packages are Bugzero [WEB05b], Census [Met05] or TestTrack Pro [Sea05]. Popular examples of the Open Source area are the bug tracking system of the mozilla project called Bugzilla [Bug05] and Eventum [MyS05a] of MySQL AB, the manufacturer of the popular MySQL [MyS05b] database server. Another, almost free, possibility is to use the SupportCenter component [Glo05] for the Mambo Open Source Content Management System. In case the Mambo system is used as a free and powerful Intranet solution the plugging in of the SupportCenter component is a very useful addition. The system is intended as a solution for customer support activities and allows to convert trouble tickets and their solution into knowledge base articles. Therefore it nicely supports the collection of knowledge arising from these efforts. A more detailed description of this solution is described in B.3.

Component: FMEA

The method of Failure Modes and Effects Analysis (FMEA) is a standard technic to assess potential product risks and failures and to evaluate on the impacts of these failures. It allows to collect and analyze per system area the different possible failures, judge on their probability, derive the associated risks and to determine remedial actions. With that assessment it becomes possible to prioritize the tackling of the different potential or already encountered problems. A more detailed introduction into the method can be found in e.g. [DRM05a]. This analysis method is often an integral part of an ISO 9000 or CMMI compliant quality system and is therefore mentioned again and described in more detail in the section 5.2.1.

A very important task when using this method is to constantly update the FMEA to be aware of the most critical issues. The whole method can be used to reduce the overall risks step by step in actively resolving the most severe items. A plentitude of different tools supporting this method exist.

4.9. Knowledge Management

4.9.1. Features provided/addressed

The building block Knowledge Management is central for an IP Engineering Framework. It adds to the features of

- Knowledge Management, by using means to collect and provide access to all necessary knowledge components and
- Reproducibility, by collecting and making the knowledge accessible again.

4.9.2. Definition

Because the Product Engineering Framework is intended for fabless, silicon based MEMS IP product development, the knowledge capture, conservation and retrieval is of special importance. Therefore the ”brick” knowledge management plays a central role and can be established by a set of tools covering at least the following aspects:

- using process-based management techniques to capture and archive experience gained during previous projects and developments
- process development design and tracking to capture systematically the generated processing knowledge and assessment results
- capturing development discussions between the engineers
- capturing customer questions and answers

For the later two standard tools can be used like knowledge management tools (e.g. [Eed05], [Adv05a]), groupware solutions e.g. [Mic05e], [IBM05b] or Customer Relationship Management tools e.g. [Ora05c], [Sie05a]). The second item is not systematically covered by commercially available tools. Because of that a novel method for capturing this part of the knowledge has been invented and will be discussed in detail in chapter 6.5 . These efforts have been integrated into the EU funded FP6 project PROMENADE [Eur04] (see also [PWOB04]) integrating the approach into a bigger framework. This project is described in much more detail in chapter 6. The first knowledge area is covered by the developed Product Management Framework discussed in chapter 3. To consider this area as a part of knowledge management is supported by [Wik05j].

Furthermore the Knowledge Management components need to provide means for capturing the design IP generated during device and circuit design activities using the developed process IP. They systematically capture the efforts using Configuration Management, set up the environments in a unified manner and so on. One important component of the development knowledge is the test data generated during experimental verification. During these verifications detailed data about devices and chips is collected for different device/chip properties. These can be electrical properties, e.g. resistances, capacitance, and so forth, mechanical properties like Youngs modulus, stress and so forth or other characterization data. All this data needs to be systematically captured in databases, related to the corresponding item and made accessible and retrievable. Additionally test tools enabling the assessment need to be present.

The knowledge management is closely related to the brick Configuration Management (CM), described in section 4.2, Change Management (see section 4.3), Requirements Engineering (see section 4.4) and Data Organization (see section 4.5). Requirements Engineering, Change - and Configuration Management keep track of all requirements, changes and consistent delivery via version control in appropriate tools. They include procedures on how requirements and change requests are defined and applied to projects in the different development stages. The Data Organization brick defines and establishes common structures to store project data, deliverables and project organizational data in a unified environment so that all staff can easily find data based on the standard categorization. All components need tight integration to facilitate reproducibility and retrieval so that the configuration management can easily be applied on the data organization. For this purpose sophisticated tools like Oracle Files [Ora05b] can be used to ease the access and search even further. Such tools allow easy collaboration over multiple sites, build a proper, easy to search reference base and support flexible access means. The build-in version management facilitates the Configuration Management. But as outlined before, theses systems are heavy and expensive.

4.9.3. Components

Component: Design setup

The general, unified design setup is described in section 4.7.3. From the knowledge management perspective it is important to integrate the whole design environment with the Configuration Management described in 4.2. One possible integration is to automatically import and checkout the working areas of the design engineers into/from the revision control system. By these means all required versions are retrievable from the revision control system. This even might include the tool versions used for the generation of a specific deliverable.

Alternatively older tools will never be uninstalled and only become archived. Overall it is essential that all tool versions remain accessible to remain reproducible.

Another component of knowledge is to know the current status. In a similar fashion as the process development status is easily made retrievable, described in 6, the design status needs to be easily retrievable as well. This can be achieved by regular builds where the results should be presented on the Intranet. Such check mechanisms have been developed in a generic fashion and are described in section 4.11.3.

Component/Brick: Configuration Management

As has been stated before, Configuration Management is one essential building block and contributor for Knowledge Management. It enables reproducibility and the systematic capture of results and therefore builds a knowledge base and archives the development progressions. Because it is described in detail in section 4.2 no further discussion is presented here again. It only needs to be restated that CM is essential for the knowledge management in the Product Engineering framework.

Component: Process Development

The process development requires special attention because it builds the main component of MEMS IP. Without a proper support for this area, especially in term of knowledge capture and development support, MEMS IP development can be a daunting and error prone task. Because of this the complete chapter 6 is dedicated to this subject. In the current context it should only be stated, that this area is essential and requires special care. The rest of the discussion is deferred.

Component: Test tools/databases

The knowledge from tests, simulation and experimental verifications is vital for the overall knowledge conservation. Therefore this data needs to be captured systematically and means must be available to easily retrieve, evaluate and assess this data. The process development tracking efforts are responsible for the systematic capture, relation and retrieval and are described in chapter 6. For the evaluation and assessment of the testing data, additional tools like statistical analysis tools or wafer mapping tools are required. They are briefly described in the following paragraphs.

Statistical analysis software To assess the testing data of devices and chips a statistical analysis software needs to be used. These software packages pro-

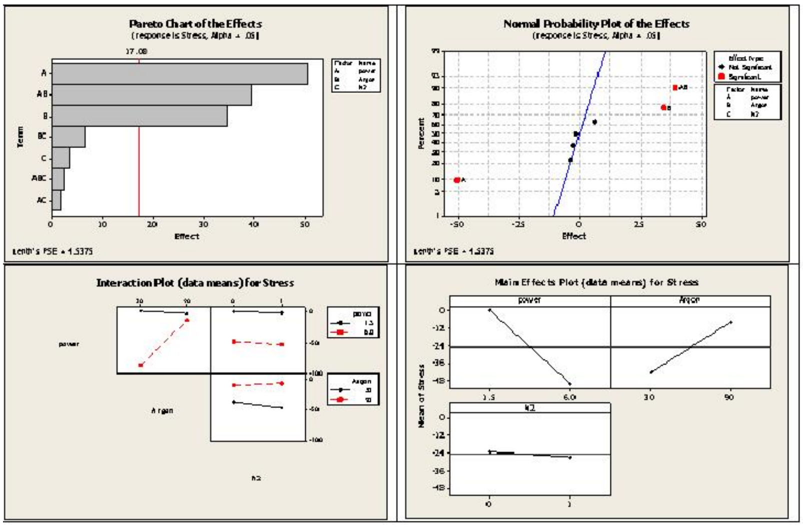


Figure 4.5.: Determination of the process parameters effect on a given property + corresponding modeling

vide comprehensive statistical and graphical analysis functionalities to perform e.g. Six Sigma analysis and other quality improvement/assessment functions. They can be used for Statistical Process Control or Design of Experiments. Examples for such tools are MiniTab [Min05], SYSTAT [Sys05a] or Statistica [Sta05c]. As an example for the possibilities of these tools figure 4.5 shows how process parameter effects can be assessed to evaluate a stress model in a cantilever structure. Another example is presented in figure 4.6 showing the value distribution of a specific test structure across a wafer and between several wafers.

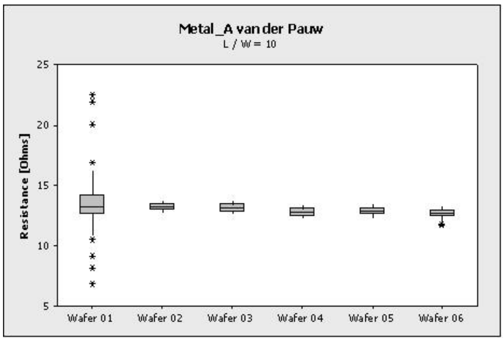


Figure 4.6.: Distribution analysis of the Van der Pauw resistance across the wafer: box plot

Excel Another component, typically available anyhow, is Excel. Although Excel is very flexible and can be used to easily present data and partly analyze data, it misses quite some features supported by the tools described above. Therefore it can only be used for certain high level task while for the real analysis the above tools are required.

Wafer mapping The statistical analysis software packages typically are application area unspecific and only provide means to analyze numerical data and present analysis results form this data. Additionally there is a need to map the numerical data to the application domain. These tools are required to allow, for example, to present data according to its origin on a wafer. Such wafer map tools, e.g. the tool Wafermap [BOI05] or ProcessNavigator [Exc05], present the testing data in a form presented similar to the one in figure 4.7. Additionally these tools typically, at least partly, support the above explained statistical analysis. Such a graphing possibility is required to more easily see deviations of one test value over the different wafer locations.

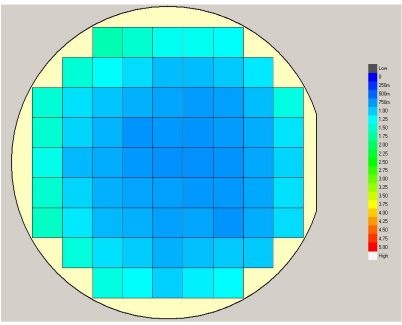


Figure 4.7.: An example conformity wafer map

Testdata database Another important development tool is the availability of a general and generic database for the wafer test results. In such a database the raw test data needs to be stored and can be retrieved and analyzed by the Statistical analysis software packages. Such a raw storage is required to be able to compare data of different wafers or lots with each other. This is due to fact that automatic probers store the results of a wafer probe in a single file and the analysis tools typical operate only a single data set. Via loading the raw data from the test files into a plain database, the excerpts of a specific analysis can easily be created or the analysis tools can access the data storage directly.

Component: Discussion Tracking

The tracking of development discussions and discussion with external sources e.g. customers are a valuable source of knowledge as well. This is especially true for questions, comments, etc. posed by an external or internal sources. They need to be answered and followed-up and provide important insight. Such questions can be considered an issue requiring resolution. This has already been discussed for the building block Issue Tracking in section 4.8. To manage the raising, coordination and resolution of such issues several approaches can be chosen as discussed in section B.3. In summary a Customer Relationship Management, a Issue Tracking System, a Document Management or an FAQ approach could be selected. All approaches have their pros and cons. From the table B.1 it can be seen that a combination of the Issue Tracking and FAQ approach would be the best fit. Therefore a system would be beneficial that supports both areas in an easy to use manner. A component supporting this is the SupportCenter component integrated into a Mambo CMS Intranet solution as described in section B.1.

Component: Intranet usage

The Intranet usage in the context of knowledge management enables to integrate and collect external and internal news. This spans from external news about technologies, competitors, market news and so forth to internal news about achievements, status and so forth. Additionally internal discussions can be captured if they are not tracked with the issues tracking approach or groupware solution. Additionally an Intranet platform offers easy mechanisms to publish, share and retrieve content. This is especially true if a Content Management System like Mambo builds the baseline for the Intranet platform.

4.10. Planning & Tracking

4.10.1. Features provided/addressed

For the project coordination and management the consequent project planning and tracking is essential. These efforts address the features:

- Target focussed, by awareness of problems and deviations,
- Control, by having the status easily accessible and
- Overall coordination, by enabling project management.

The importance of Planning and Tracking is emphasized by the Product Engineering Method as well. Here the planning processes are required by several other processes of the method.

4.10.2. Definition

Project planning and tracking is the central enabler of project management. A good introduction into the subject of project planning is given in [Gid85]. Without constantly planning and tracking the project progress, including necessary replanning and rescheduling, it is impossible to have control. Therefore these efforts are the key point to enable control and need to maintain a project plan and regularly update the completion status. Having control is essential to be able to detect deviations and define and execute contingency plans. These can ensure that project goals are reached although unforseen events were encountered. All planning and tracking efforts need to acknowledge project planning best practices as described in e.g. [IBM05e], [Mar05]. The later one gives a good comparison about different project management approaches and methodologies. Additionally this citation is a valuable resource with further links on the subjects project planning and management. To summarise the most important requirements for a good plan it can be said that the plan must contain (obvious but worthwhile mentioning) a chronological listing of products or deliverables broken down into activities, actions and tasks including the dependencies between the different deliverables. A plan requires a timeline and schedule, showing beginning and end points of the actions (aggressive but realistic); must state the resources required for each action or task, notably persons, person-days and money; must contain milestones to be achieved meanwhile.

To perform the overall organizational coordination, tool support for the project management according to PRINCE2 is required. The PRINCE2 standard already contains quite a few predefined checklists and templates to coordinate diverse activities but additional tools are required. These tools need to cover the support of the planning and tracking itself. Additionally a constant presentation of the current project status is preferable to allow early noticing of deviations and easy status communication. Further, more organizational components, need to cover easy coordination and easy communication, described in section 4.7 about the Common Environment.

4.10.3. Components

Component: Plan and Track

The Planning and Tracking efforts of **all** projects is essential as outlined above. To support the efforts an appropriate planning and tracking tool is required. One of the standard tools for this purposes is Microsoft Project [Mic05f]. Hundreds of alternative solutions like MJI Teamworks [MJI05] or Project Kickstart [Exp05] exist. MS Project has its pros and cons but can be considered the standard for planning efforts. Considering the exchange with external partners it is a sensible choice. To enable reproducibility, the project plans need to be kept in the standard electronic binder of the project data organization (4.5) and have to be under revision control to keep the history for future reference.

Component: Constant status presentation

The constant presentation of the project status enables that all involved parties are aware of deviations and project slippage. Only this awareness enables control and the definition and execution of contingency plans. Therefore a component showing the current status in an easy to understand overview is very useful. Such a component should be integrated into the Intranet environment and give a highlighted visibility with reduced complexity only. Such a component is described in more detail in section 4.12.3.

4.11. Regular Execution

4.11.1. Features provided/addressed

The regular execution of the development processes and their sub tasks addresses the PEF features of:

- Easy and efficient communication, by regularly talking about repeating topics,
- Target focused, by awareness of problems,
- Control, by having problems visible and
- Efficiency, by executing along same lines.

4.11.2. Definition

The regular execution of the development process and its sub processes is necessary to keep the quality levels, comply with the CMMI[2] level and keep the engineering staff on the same track. To reach at least CMMI level 3 (Defined) the regularity is mandatory. The regular execution of the processes fosters that communication becomes easier, several tasks become self-evident, project management and control is established and made easier and the overall development therefore becomes more efficient. The execution of the processes is established and enforced by the quality assurance.

To reduce the overhead, some tasks of regular execution can be automated. This ranges from regular updates of work spaces from the repositories to a generic building mechanism recreating the full project deliverables on a regular basis and publishing the results on the Intranet. Such an approach has been introduced in [DBG+03] and has proven to be very useful and efficient. This is due to early detection of issues which can be tackled accordingly.

[2] CMMI stands for Capability Maturity Model Integrated and is described in more detail in e.g. [CKS03], [Wik05a].

4.11.3. Components

Component: Process execution

The development methodology and the general usage of all building blocks fosters the regular execution of the different sub-processes. This is because all development projects, ranging from small to very big, are all executed along the same processes and procedures. The only difference between the different size projects is that smaller projects can opt to execute along slimmer process versions. By that they can reduce the management overhead in reducing the formal steps to the compulsory subset. The (sub-)processes and procedures are the crucial mortar between the different bricks. They use the tools and procedures of the bricks to move the development efforts along, following the guidelines and processes of the methodology. The compliance with this scheme is enforced by the quality gates in the development processes and by the additional quality assurance measures like project audits and project management audits.

Component: Nightly build

For the product engineering framework means are necessary to regularly update the project directories with the data from the Configuration Management. Additional mechanisms to regularly generate a project status update in the form of a detailed deliverable compilation and status check are instrumental. While the first feature is necessary for every deliverable of a project, the second feature is special for design and software deliverables. This is because automatic checking of all kinds of deliverables is not possible e.g. to check a Word document for its completion. Therefore a build and project update environment is required to enable early issue detection.

The generally established build mechanism is using the project specific build setups to execute a regular and complete rebuild of the project. To remain flexible and generic, the responsibility to define the necessary build tasks is delegated to each project and its tool smith while the frame for the execution, analysis and presentation is provided by the general environment. This building is executed at least once a day, typically during the night, using the distributed and clustered environment. The results are analyzed by the build result evaluation tools and are presented in a unified way on the Intranet. A more detailed description of the implemented build and execution environment is given in section B.4.

4.12. Regular Status

4.12.1. Features provided/addressed

The regular generation, update and publishing of the current project status addresses the following features of the Product Engineering Framework:

- Easy and efficient communication, by having the status visible,
- Target focused, by awareness of problems and
- Control, by having status easily accessible.

This adds to the possibilities to detect issues early, to easily be and make the audience aware of the status and to highlight slippage.

4.12.2. Definition

Building a regular status of a development project is essential to be able to manage and have control of such efforts. Only via the current status it is possible to determine the necessity to develop or execute contingency plans. Only these remedial actions can keep a project on track once it starts slipping or other hurdles are encountered. Therefore the building of a regular status is an essential vehicle.

To regularly build a status, several components can be used like regular status reports, status update meetings, regular project builds and change monitoring. All components add to the picture of the current status and help identifying deviations and early detection of issues.

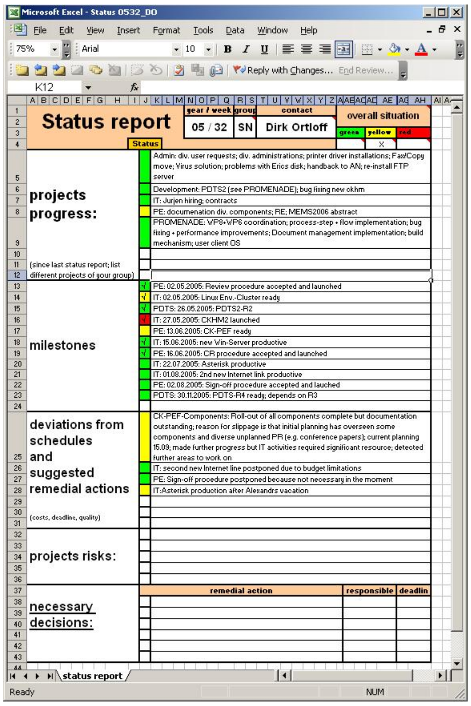
4.12.3. Components

Component: Status reports

A status report is supposed to briefly summarise the status of a block of activities and/or deliverables. Because the report only evaluates on the status, it does not provide detailed technical information. It only provides management with the most essential achievements since the last report, gives an overview about the projects milestones, evaluates the deviations from the schedule and remedial actions, highlights the project risks and indicates necessary decisions. To ease the viewing and detecting of the problematic areas a traffic light system is used to indicate the criticality of each item and give an impression of the overall project/group status. An example of a status report is presented in figure 4.8.

Component: Regular status update meeting

Although meetings are often considered unnecessary overhead, a regular project status update meeting is required to put everybody onto the same page. Additionally these meeting build the forum to update the project plan with completion percentages, enable discussion, define contingency plans and assess the necessity to execute contingency plans. These meetings should involve the complete project staff to prevent information lacks due to inappropriate distribution. To support these meetings, the basic components conference calls and online conferences of the common environment (refer to 4.7) should be used, especially if staff is distributed over several sites. As for all meetings, the agenda needs to be published in advance and minutes need to be taken, distributed and archived for reference in the standard electronic binder of the project data organization (4.5).

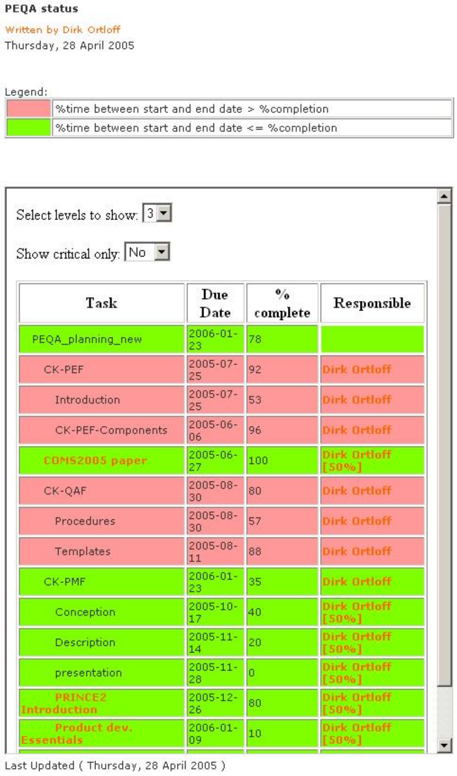

Microsoft Excel - Status 0532_DO
File Edit View Insert Format Tools Data Window Help
75%
Arial
10
B I U
Reply with Changes... End Review...
K12
status report
Ready
NUM

Status report	year / week	group	contact	overall situation
	05 / 32	SN	Dirk Ortloff	green: yellow: X red:

	Status	
projects progress:	🟩	Admin: div. user requests; div. administrations; printer driver installations; Fax/Copy move; Virus solution; problems with Erics disk; handback to AN; re-install FTP server
	🟩	Development: PDTS2 (see PROMENADE); bug fixing new ckhm
	🟩	IT: Jurjen hiring; contracts
	🟨	PE: documenation div. components; RE; MEMS2006 abstract
	🟩	PROMENADE: WP8+WP6 coordination; process-step + flow implementation; bug fixing + performance improvements; Document management implementation; build mechanism; user client OS
(since last status report; list different projects of your group)		
milestones	🟩 √	PE: 02.05.2005: Review procedure accepted and launched
	🟨 √	IT: 02.05.2005: Linux Env.-Cluster ready
	🟩 √	PDTS: 26.05.2005: PDTS2-R2
	🟥 √	IT: 27.05.2005: CKHM2 launched
	🟨	PE: 13.06.2005: CK-PEF ready
	🟩 √	IT: 15.06.2005: new Win-Server productive
	🟩 √	PE: 16.06.2005: CR procedure accepted and launched
	🟩	IT: 22.07.2005: Asterisk productive
	🟩	IT: 01.08.2005: 2nd new Internet link productive
	🟩	PE: 02.08.2005: Sign-off procedure accepted and lauched
	🟩	PDTS: 30.11.2005: PDTS-R4 ready; depends on R3
deviations from schedules and suggested remedial actions	🟨	CK-PEF-Components: Roll-out of all components complete but documentation outstanding; reason for slippage is that initial planning has overseen some components and diverse unplanned PR (e.g. conference papers); current planning 15.09; made further progress but IT activities required significant resource; detected further areas to work on
	🟩	IT: second new Internet line postponed due to budget limitations
	🟩	PE: Sign-off procedure postponed because not necessary in the moment
	🟨	IT:Asterisk production after Alexandrs vacation
(costs, deadline, quality)		
projects risks:		

necessary decisions:	remedial action	responsible	deadlin

Figure 4.8.: Example of a project status report



PEQA status

Written by Dirk Ortloff
Thursday, 28 April 2005

Legend:

🟥	%time between start and end date > %completion
🟩	%time between start and end date <= %completion

Select levels to show: 3

Show critical only: No

Task	Due Date	% complete	Responsible
PEQA_planning_new	2006-01-23	78	
CK-PEF	2005-07-25	92	Dirk Ortloff
Introduction	2005-07-25	53	Dirk Ortloff
CK-PEF-Components	2005-06-06	96	Dirk Ortloff
COMS2005 paper	2005-06-27	100	Dirk Ortloff [50%]
CK-QAF	2005-08-30	80	Dirk Ortloff
Procedures	2005-08-30	57	Dirk Ortloff
Templates	2005-08-11	88	Dirk Ortloff
CK-PMF	2006-01-23	35	Dirk Ortloff
Conception	2005-10-17	40	Dirk Ortloff [50%]
Description	2005-11-14	20	Dirk Ortloff [50%]
presentation	2005-11-28	0	Dirk Ortloff [50%]
PRINCE2 Introduction	2005-12-26	80	Dirk Ortloff [50%]
Product dev. Essentials	2006-01-09	10	Dirk Ortloff [50%]

Last Updated (Thursday, 28 April 2005)

Figure 4.9.: Example of a project status presentation on the Intranet

Component: Regular status presentation

The idea of the regular project status presentation is to make the project status and the status of all deliverables easily viewable via an Intranet page and limit the presented data to absolute minimum. To limit the extra effort to create this presentation, a solution was set up retrieving the status from the MS Project plan. The update of this plan is implied by the development method anyhow and can deliver all required data. For that all deliverables should be represented as a task/milestone in the planning as required by the product based planning technique. Their status needs to be kept up to date and via adding hyperlinks to the tasks it is possible to link directly to the deliverables (orange name in figure 4.9). These should reside in the revision control system (section 4.2) and using CVSweb they can directly be retrieved by a single click. Additionally it is possible to directly contact the person responsible for a deliverable. The resources of a plans task enable to add direct email addresses to the tasks. They are shown as links (orange labels in figure 4.9) opening an emailing window with pre-configured mail address and subject. To ease the viewing even further, colouring and detail selection features are implemented. This provides means to get an impression of the project status by a single glance. An example for the presentation is given in figure 4.9.

Component: Nightly build

A regular deliverable status check is performed by the nightly build processes, introduced in section 4.11.3. But this is only possible for design or software deliverables because the automatic completion checking or rebuild of e.g. Word documents is difficult to achieve. For these deliverables the above described Regular status presentation on the Intranet from the the MS Project plan needs to be used. The nightly build produces a detailed, project configurable status overview of all deliverables which can be automatically build and checked. A more detailed introduction of the approach can be found in section 4.11.3 and its base paper [DBG+03].

Component/Brick: Change monitoring & presentation

Another component for a regular status update and presentation is the monitoring of changes in the repository and the appropriate presentation of these changes. This method can be used to steadily be informed about changes done to the different projects and its deliverables. Additionally it can be used as an entry point for questions like: "Which items have been changed when?". The change monitoring produces a presentation of the changes over the timescale on different levels (repository, module). A more detailed introduction into the change monitoring and used tool components is given in section 4.2.

4.13. Enforcement

4.13.1. Features provided/addressed

The building block enforcement adds to the following features of the Product Engineering Framework

- Control, by knowing which deliverables are minimal required and
- Overall coordination, by providing guidelines for the daily work.

4.13.2. Definition

With the Enforcement of the organizations methods, procedures and policies it is possible to gain more control and raise the quality level. By enforcing that certain steps need to be taken within a development project via the gate assessments, the tendency to cut corners is reduced and the work becomes more efficient. This is due to the fact that using standard procedures requiring certain steps to be taken or deliverables to be produced the repeated flow becomes faster and a guideline for the different project situations is provided.

The Enforcement is achieved by an active quality assurance system, by predefined, easy to execute procedures and the gating development method.

4.13.3. Components

Active Quality Assurance

An active quality assurance involves itself into the development from the project start to its end. The intention is not to burden the project. Instead orientation in the project jungle, method support and active review possibilities are provided. Therefore the active quality assurance guides the developments by providing tools, guidelines and standards to the project. By insisting on certain project deliverables, the quality assurance procedures and means add to the features of reproducibility and enforce a quality fashion execution of the development efforts. A more detailed description of the quality assurance framework is given in section 5.

Procedures

On the one hand the usage of fixed procedures causes certain overhead. On the other hand this overhead is required to enforce a quality fashion execution. The procedures, like the deliverable review procedure or change request procedure, provide help getting to agreements in certain project situations or for certain project matters. The procedures enforce specific steps to be taken to prevent cutting corners. By that they ensure high quality deliverables and help to increase the projects potential success rate.

Gating development methodology

The gating development methodology introduced in section 3.4.3 enforces that certain synchronization points in the project need to be used. During these points conscious decisions about the further project progress will be taken, including the necessary management attention at these points. Due to that the methodology defines a certain hurdle hight for the gates and enforces a certain quality of the deliverables at the gate including an independent cross check.

4.14. Vision

4.14.1. Features provided/addressed

The building block Vision provides for the following features of the Product Engineering Framework:

- Easy and efficient communication, by common vocabulary and unified aim,
- Target focused, by communication and early agreed expectations and
- Control, by setting everybody onto the same page.

In the general sense it is rather difficult to create a vision for an organization or project. But as outlined in [NSB05] this is rather essential for success because it is an important part of the team and culture building. The article gives an outline of how to create a vision and the necessary steps as well. Another resource with advice for this subject is [Adv05b]. Additionally plenty of books reflect the subject ”Creating a Vision and ”Forward Thinking”.

4.14.2. Definition

The brick Vision is crucial for the overall success of a development project. It is achieved by an early product definition developed during the first activities of the product development methodology (refer to 3.4). It enables to clearly define and fix the goal of the developments (to prevent shooting onto moving targets). This vision can only be changed using the formalized Change Request procedure. This includes management approval during the executing of the assessments of a development gate.

Several components have to be used to create an appropriate Vision. This includes the requirements specification and the early product definition and Project Brief of the PRINCE2 method (see definitions below). All the project and product deliverables together shape up a the Vision of a project.

As a matter of fact the sequence of the creation of these deliverables can vary depending on the product type. While for a market driven product, where the market need is the first initiator, the requirement specification is typically the first deliverable. In these projects the vision statement or product brief is then typically the second deliverable and the product definition is the third deliverable. For an idea driven product the first two deliverables are typically the other way round. In this case the requirement specification is used to correct the product brief in measuring the vision statement against the market needs.

4.14.3. Components

A vision can be created by the following components becoming deliverables of a development project.

Component: Requirements specification

The requirements specification can be seen as the request from marketing&sales to the development organization defining the market needs and the boundary constraints of a potential product. Depending on the product type this specification is used as a product development initiator or as a correcting vehicle for the product vision statement for an idea driven project. The requirement specification is described in more detail in section A.3.3.

Component: Vision statement or Product Brief

The vision statement or Product Brief is usually created during the set up of a development project as part of the project startup activities. It defines the project scope in terms of the development aim. For a market driven product the vision statement clarifies which requirements of the Requirement Specification will be addressed by the product and which are out of the scope of the project. It clearly defines what should be achieved by the project and gives an overview of the operation domain. The scope of the Vision statement is described in more detail in section A.3.4.

Component: Product Definition

The Product Definition is a kind of an answer of the development organization to the Product Brief and Requirement Specification. It is developed after the initial concept assessments and prototyping tests are performed and defines in detail the characteristics and limitations of the product. Therefore it gives an realistic picture of the outcome of the development project in the light of the other two vision building components. The contents of the product definition is described in more detail in section A.3.5.

4.15. Processes/Procedures/Mortar

The mortar between all building blocks is composed out of the processes and procedures of the Project Management Framework and the Quality Assurance Framework. The development methodology and the different development processes and sub-processes are set up in a way that all development projects, ranging from small to very big projects, are all executed along the same processes and procedures. The only difference between the different size projects is that smaller projects can opt to execute along slimmer process versions and can reduce the management overhead by reducing the formal steps to the compulsory subset. The (sub-)processes and procedures are the crucial mortar between the different bricks using the tool components and procedures of the building blocks to move the development efforts along, following the guidelines and processes of the methodology. For example the project management process ″Planning″ uses the ″Planning″ brick as well as e.g. the ″Regular Status″, ″Config. Management″, ″Common environment″ bricks. Another example is the ″Change request procedure″ of the quality framework using e.g. the ″Config. Management″, ″Change Management″, ″Requirements Engineering″. Because the processes and procedures are already described in sections 3 and 5 no further discussion is given in this section.

Additionally the integration of the tools of the different bricks can be considered mortar as well. As has been presented in chapter 4 several tool components are mentioned within several building blocks. Therefore they build themselves a kind of mortar between the bricks. Additionally all tools are integrated into the common environment and most are integrated via the Intranet. Therefore they are connected to each other by these means. On top of that, by using several conventions, further integration can be achieved. One example is the integration between ″Issue tracking″ and ″Config. Management″. If using CVS and e.g. Bugzilla putting special keywords into the check-in comments for CVS e.g. ″BUG XXX″ a direct link into Bugzilla can be created using CVSmonitor. This enables that CVSmonitor can be used to refer from a CVSmonitor ChangeSet fixing a specific bug directly into the related Bugzilla bug. Another example is the integration of the bricks Configuration Management and Data Organization. These are integrated by having all folders of the electronic project binder as separate modules in the project repository of the Configuration Management, as described in section 4.5.

4.16. Conclusions

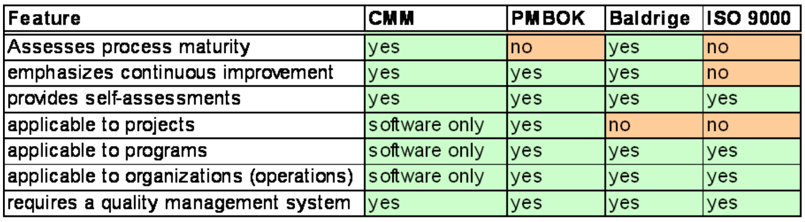
This section introduced an Engineering Framework, its required features and tool components, software and process tools, providing these features. For that certain areas to be covered have been identified, these areas have been defined and appropriate components fulfilling the needs have been introduced and discussed. This unique and novel assembly of the different components supports the newly developed product development methodology introduced in section 3. They build a round trip framework for engineering silicon based MEMS IP products.

5. Quality Assurance Framework

The Quality Assurance Framework adds additional components to the overall development framework supporting and ensuring high quality development results. These are additions to the quality measures already build into the product engineering method and product engineering framework. The current chapter briefly summarizes components and tools useful to support and ensure quality development. The components and procedures range from very basic ones like deliverable review procedures and change request procedures to higher level components. The higher level components are project audit and project management audit procedures. To introduce the background for these additions, the first section will briefly summarize general quality framework and improvement approaches.

5.1. Quality & Quality Assurance approaches

Before going into the details of different quality approaches, a short comparison between different, generally applicable approaches should be provided. The approaches under consideration are the ISO 9000, CMMI and the Baldrige method. They have been compared very nicely in [PSI05]. A more comprehensive comparison is given in [Com04] and a rating of their generality is presented in figure 5.2. In summary it can be said that ISO 9000 is an effective framework for assessing the minimum requirements and processes for a quality management system. It is especially useful if a certification is valuable to the organization. CMMI (in [PSI05] CMM, the predecessor of CMMI) is an effective framework for modelling, defining, and assessing the maturity of the processes used within projects, programs and operations (business unit or enterprise). Additionally it identifies the key practices that are required to increase the maturity of these processes. The Baldrige approach focusses on implementing & assessing the maturity of all processes used within all enterprise operations (business unit level or enterprise level). A comparing summary of these approaches is given in figure 5.1, extracted from the above citation. The figure compares key features of several different quality approaches as well as the project management method PMBOK. The comparison only lists CMM intended to be used as a maturity model for software developments. CMMI is a successor of CMM extended to be applicable in several different industry



Feature	CMM	PMBOK	Baldrige	ISO 9000
Assesses process maturity	yes	no	yes	no
emphasizes continuous improvement	yes	yes	yes	no
provides self-assessments	yes	yes	yes	yes
applicable to projects	software only	yes	no	no
applicable to programs	software only	yes	yes	yes
applicable to organizations (operations)	software only	yes	yes	yes
requires a quality management system	yes	yes	yes	yes

Figure 5.1.: Comparison between different quality concepts (extracted from [PSI05])

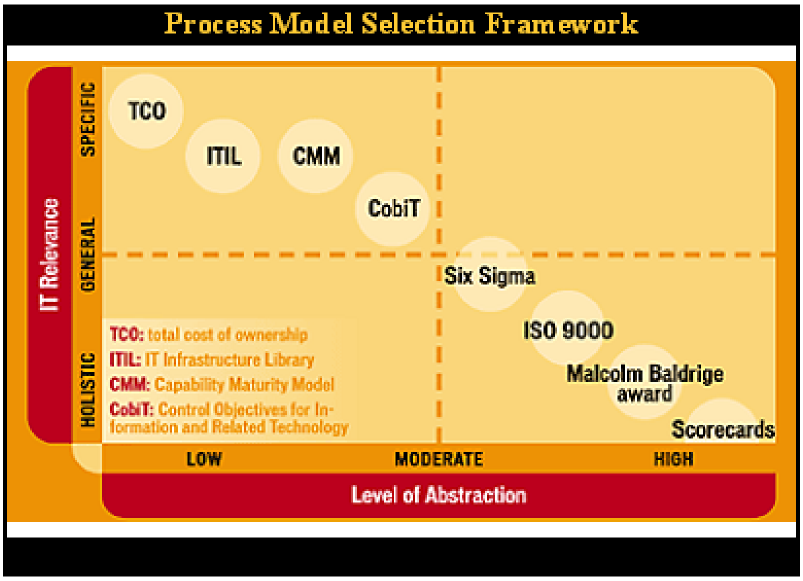

Process Model Selection Framework
IT Relevance
SPECIFIC
GENERAL
HOLISTIC
TCO
ITIL
CMM
CobiT
Six Sigma
ISO 9000
Malcolm Baldrige award
Scorecards
TCO: total cost of ownership
ITIL: IT Infrastructure Library
CMM: Capability Maturity Model
CobiT: Control Objectives for Information and Related Technology
LOW
MODERATE
HIGH
Level of Abstraction

Figure 5.2.: Process model selection framework (extracted from [Com04])

areas. In that respect the fields filled with "software only" can be ignored for the CMMI approach.

5.1.1. ISO9000 family

The ISO 9000 standards define minimum requirements for business quality assurance systems. It is a family of different standards covering different application domains and setting minimum requirements for the quality assurance systems. They are maintained by the International Standards Organization (ISO) in Europe and the American National Standards Institute (ANSI) in the United States. They have been released in 1987, 1994 and 2000. Usually the conformance is voluntary. However, many European firms use them as a requirement for suppliers. Within the United States, some firms also use the standards for supplier certification.

The ISO does not issue certifications. For that assessors are used which themselves need to be approved by an accreditation body (e.g. Loyds Register, http://www.lr.org). In the United States the American Society for Quality Control has organized a Registrar Accreditation Board (RAB). This board oversees the training and activities of assessors and certify conformance with the requirements. Certification is evidence to customers, potential customers and others that a business meets the standard's requirements.

Assessors typically spend 1-3 days at a site. They examine documents, interview employees and observe processes. They look for evidence to confirm compliance or non-compliance. To obtain certification, a business may contract with an accredited assessor. An important customer may wish to participate in the assessment as part of their supplier certification. Management may wish a self-assessment for internal evaluation.

As mentioned above, the ISO 9000 consists of a series of standards which are:

ISO 9000 is a guideline for selection and use of quality system standards. It provides insight for various situations and conditions as well as definitions and explanations.

ISO 9001 defines minimum quality system requirements for design/development, production, installation and servicing. It is the most complete standard. It applies to manufacturing and service businesses engaged in all these activities.

ISO 9002 is essentially a subset of 9001. It applies only to production and installation activities.

ISO 9003 applies to final inspection and test.

ISO 9004 is a guideline for quality system elements. It is like a textbook which describes, explains and recommends.

Since the ISO 9000:2000 revision the ISO 9002 and ISO 9003 have been discontinued and are substituted by the ISO 9001:2000.

All standards cover the areas of Management responsibility, Quality System, Contract Review, Design Control, Doc/Data control, Purchasing Control of customer supplied product, Product ID & traceability, Process Control, Inspection & testing, Control of inspections, Inspection & Test Status, Control of Nonconforming Product, Corrective & Preventive Action Handling, Storage, Packaging, Delivery, Control of quality records, Internal Quality Audits, Training, Servicing, and Statistical Techniques.

A more detailed introduction into the ISO 9000:2000 can be found in [Gro06].

5.1.2. CMMI

Besides ISO 9000, the Capability Maturity Model Integration (CMMI) developed by the Software Engineering Institute (SEI) of Carnegie Mellon University is gaining increasing importance in Europe and around the world.

CMMI is a process model. In contrast to a concrete process description, CMMI defines demands on an efficient product development process, i.e. what is necessary. However, it does not define the concrete steps that must be taken to reach an efficient development process. The primary goal of CMMI is to support a continuous process improvement, by defining demands, respectively criteria, of a professional product development organization.

CMMI can be used to guide process improvements across a project, a division, or an entire organization. The model describes requirements for established development standards and groups them into maturity levels. CMMI differentiates five steps of maturity levels that a development process in a company can have: Initial (Level 1), Managed (Level 1), Defined (Level 3), Quantitatively Managed (Level 4) and Optimizing (Level 5). For each level key process areas have been defined which need to be established to reach that level, e.g. change management that is executed by an organization at level 2. The different process areas required for the different levels are summarized in table 5.1. CMMI also defines precise process requirements for all processes on each level. Through ascending to a higher level, a company may significantly improve its development processes, the product quality and the financial position. The approach is described in more detail in [CKS03] and [Wik05a] and its homepage is [Car05].

CMMI can be applied in order to objectively analyze the strengths and weaknesses of a product development process. It allows to decide on improvement measures and to realize them step-by-step according to their degree of logic and importance. Primarily, CMMI is a tool to improve the product development processes. Secondarily it functions as an official assessment of the maturity level of the development process. A certification of the level, which is de facto acknowledged in industry, can be obtained as well.

Abbr.	Name	Maturity Level
CM	Configuration Management	2
MA	Measurement and Analysis	2
PMC	Project Monitoring and Control	2
PP	Project Planning	2
PPQA	Process and Product Quality Assurance	2
REQM	Requirements Management	2
SAM	Supplier Management Agreement	2
DAR	Decision Analysis and Resolution	3
IPM	Integrated Project Management	3
ISM	Integrated Supplier Management	3
IT	Integrated Teaming	3
OEI	Organisational Environment for Integration	3
OPD	Organisational Process Definition	3
OPF	Organisational Process Focus	3
OT	Organisational Training	3
PI	Product Integration	3
RD	Requirements Development	3
RSKM	Risk Management	3
TS	Technical Solution	3

Abbr.	Name	Maturity Level
VAL	Validation	3
VER	Verification	3
QPM	Quantitative Project Management	4
OPP	Organisational Process Performance	4
CAR	Causal Analysis and Resolution	5
OID	Organisational Innovation and Deployment	5

Table 5.1.: CMMI processes and maturity levels

5.1.3. Malcom Baldrige Criteria

Another set of quality criteria is the Malcom Badrige Criteria catalogue. The criteria and an corresponding award is named after Mr. Malcolm Baldrige, who served as US Secretary of Commerce from 1981-1987, and whose managerial excellence contributed to long-term improvement in efficiency and effectiveness of the US government. The Malcolm Baldrige National Quality Award (MBNQA) is the premier award for business excellence and quality achievement in the United States. Since 1988, 49 organizations have received this recognition for demonstrating ever-increasing value to their customers, organizational effectiveness and learning. The award promotes quality awareness, recognizes the quality and performance achievements of U.S. organizations, and publicizes successful performance strategies which ensure business excellence. It has five categories: manufacturing, service, small business, education and health care. The award is not given for specific products or services.

The MBNQA is given to U.S. organizations that have exemplary achievements in the seven areas described below. These seven areas affect all key stakeholders, including: communities, customers, employees, shareholders and suppliers. All applicants for the Baldrige Award undergo a rigorous examination process that ranges from 300 to 1,000 hours of outside review. All applications are reviewed by an independent board of examiners primarily from the private sector. Each applicant receives a report citing strengths and opportunities for improvement.

Along the way, thousands of companies have used the Baldrige criteria to assess performance and measure their progress relative to a recognized compilation of best practices.

The seven categories known as the Baldrige Criteria:

1. Leadership – The Leadership Category examines how the organization's senior leaders address values, directions and performance expectations. Further there is a focus on customers and other stakeholders, empowerment, innovation, and learning. Also examined are the organization's governance and how the organization addresses its public and community responsibilities.

2. Strategic Planning – The StrategicPlanning Category examines how the organization develops strategic objectives and action plans. It is also examined how chosen strategic objectives and action plans are deployed and how the progress is measured.

3. Customer and Market Focus – The Customer and Market Focus Category examines how the organization determines requirements, expectations and preferences of customers and markets. Also examined is how the organization builds relationships with customers and determines the key factors that lead to customer acquisition, satisfaction and retention and to business expansion.

4. Measurement, Analysis, and Knowledge Management – This category examines how the organization selects, gathers, analyzes, manages, and improves its data, information and knowledge assets.

5. Human Resource Focus – The Human Resource Focus Category examines how the organization's work systems and employee learning and motivation enables employees to develop and utilize their full potential in alignment with the organization's overall objectives and action plans. Also examined are the organization's efforts to build and maintain a work environment and an employee support climate conducive to perform excellence and to personal and organizational growth.

6. Process Management – The Process Management Category examines the key aspects of the organization's process management. This includes the key product, service, and business processes for creating customer and organizational value and their key support processes. This Category encompasses all key processes and all work units.

7. Business Results – This category examines the organization's performance and improvement in key business areas - customer satisfaction, product and service performance, financial and marketplace performance, human resource results, operational performance and governance and social responsibility. Also examined are performance levels relative to those of competitors.

5.1.4. Conclusions

Because the main focus of the current evaluation are the quality assurance aspects of the development environment and process, the application of the Baldrige criteria are a bit too comprehensive. Therefore only CMMI and ISO 9000 should be considered for setting the quality measures. CMMI and the ISO 9000 series of standards share common concerns with quality and process management. While CMMI emphasizes continuous improvement, ISO deals with minimum criteria of quality systems. The implications are that an ISO 9001-compliant organization would not necessarily satisfy all criteria of the CMMI level 3 key process areas (it would satisfy most of the level 2 goals and many level 3 goals). Some key practices in CMMI are not addressed in ISO 9000, e.g. it is possible for a level 1 organization to receive an ISO 9001 registration; similarly, there are areas addressed by ISO 9001 that are not addressed in the CMMI. However a CMMI level 3 organization has little difficulty in obtaining ISO 9001 certification.

With the constant application of the development methodology described in section 3.4.3 and its integrated PRINCE2 method, all principle criteria for reaching CMMI level 3 are fulfilled. This point of view is shared by [ope06]. Therefore the quality efforts should focus on the application of the CMMI criteria and apply the constant process improvements to reach level 3. With achieving this maturity level 3 an ISO 9001 certification is in reach for companies applying the developed methodology.

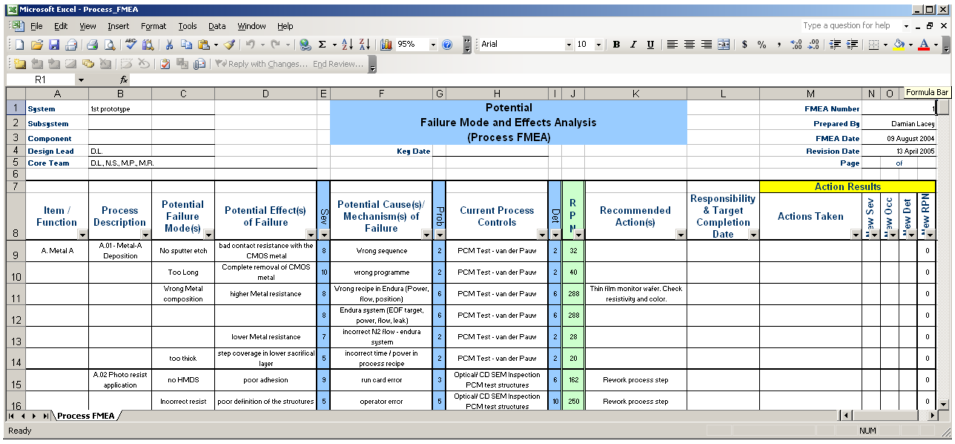
5.2. Quality components and tools

The current section gives a brief overview of the quality components and tools foreseen for the environment. A very comprehensive collection of quality related standards, techniques, etc. is given in the book Quality Essentials [ReV04]. Future extensions of this initial quality assurance set up might integrate additional measures described in this book.

5.2.1. FMEA

The Failure Modes and Effects Analysis (FMEA) technique is a standard to assess potential product risks and failures. It is used to evaluate on the impacts of these failures. It allows to collect and analyze different possible failures per system area, judge on their probability, derive the associated risks and to determine remedial actions. A more detailed introduction into the method can be found in e.g. [DRM05a].

The method typically generates a list of issues/potential issues and then systematically derives the other important aspects like potential impact of the issues/failure, effects of the failure mode, detection controls, remedial actions and so forth step by step in a spreadsheet manner. An example of such a matrix is given in figure 5.3. The figure shows the most important columns of severity (labelled as ”Sev”), probability (labelled as ”Prob”) and detection probability (labelled as ”Det”) assigned to each failure with blue color. For each of these three categories value intervals need to be defined for the whole FMEA and each failure mode receives one value from this interval. Additionally the method determines the resulting value for the Risk Priority Number



System 1st prototype
Subsystem
Component
Design Lead D.L.
Core Team D.L., N.S., M.P., M.R.

Potential
Failure Mode and Effects Analysis
(Process FMEA)

Key Date

FMEA Number 1
Prepared By Damian Lacey
FMEA Date 09 August 2004
Revision Date 13 April 2005
Page of

Item / Function	Process Description	Potential Failure Mode(s)	Potential Effect(s) of Failure	Sev	Potential Cause(s)/ Mechanism(s) of Failure	Prob	Current Process Controls	Det	RPN	Recommended Action(s)	Responsibility & Target Completion Date	Action Results				
												Actions Taken	New Sev	New Occ	New Det	New RPN
A. Metal A	A.01 - Metal-A Deposition	No sputter etch	bad contact resistance with the CMOS metal	8	Wrong sequence	2	PCM Test - van der Pauw	2	32							0
		Too Long	Complete removal of CMOS metal	10	wrong programme	2	PCM Test - van der Pauw	2	40							0
		Wrong Metal composition	higher Metal resistance	8	Wrong recipe in Endura (Power, flow, position)	6	PCM Test - van der Pauw	6	288	Thin film monitor wafer. Check resistivity and color.						0
				8	Endura system (EOF target, power, flow, leak)	6	PCM Test - van der Pauw	6	288							0
			lower Metal resistance	7	incorrect N2 flow - endura system	2	PCM Test - van der Pauw	2	28							0
		too thick	step coverage in lower sacrifical layer	5	incorrect time / power in process recipe	2	PCM Test - van der Pauw	2	20							0
	A.02 Photo resist application	no HMDS	poor adhesion	9	run card error	3	Optical/ CD SEM Inspection PCM test structures	6	162	Rework process step						0
		Incorrect resist	poor definition of the structures	5	operator error	5	Optical/ CD SEM Inspection PCM test structures	10	250	Rework process step						0

Figure 5.3.: An example FMEA assessment matrix

per failure mode by multiplying the the previous three values in the next green highlighted column labelled as ”RPN”. The resulting ”RPN” values of all the different potential failures determine the priority of the different potential failures and can be used to assess and realize the remedial actions defined in the last columns.

A very important task when using this method is to constantly update the FMEA to be aware of the most critical issues. The whole method can be used to reduce the overall risks step by step in actively resolving the most severe items.

5.2.2. Change Request procedure

The Change Request procedure defines the roles, the essential components and the procedure itself required to modify officially released deliverables or deliverables under development. It has the purpose to give the engineering and quality framework predefined, easy to execute means to achieve controlled change of already internally or externally released deliverables or to change products currently under development. Because of that it is one component of Change Management and creates the gateway to propose, evaluate and execute requested changes. The Change Management (see section 4.3), together with the Configuration Management (see section 4.2) and the Requirements Management (see section 4.4), takes care of keeping the CRs and the different revisions as well as the requirements for a product in synchronisation.

The change request procedure provides a framework to efficiently manage the change requests and apply them to the targeted deliverable(s) or project. The process distinguishes three variants depending in the implications of the requested change. For changes doable within the given project and stage constraints, the change will be worked into the stage planning and executed. If

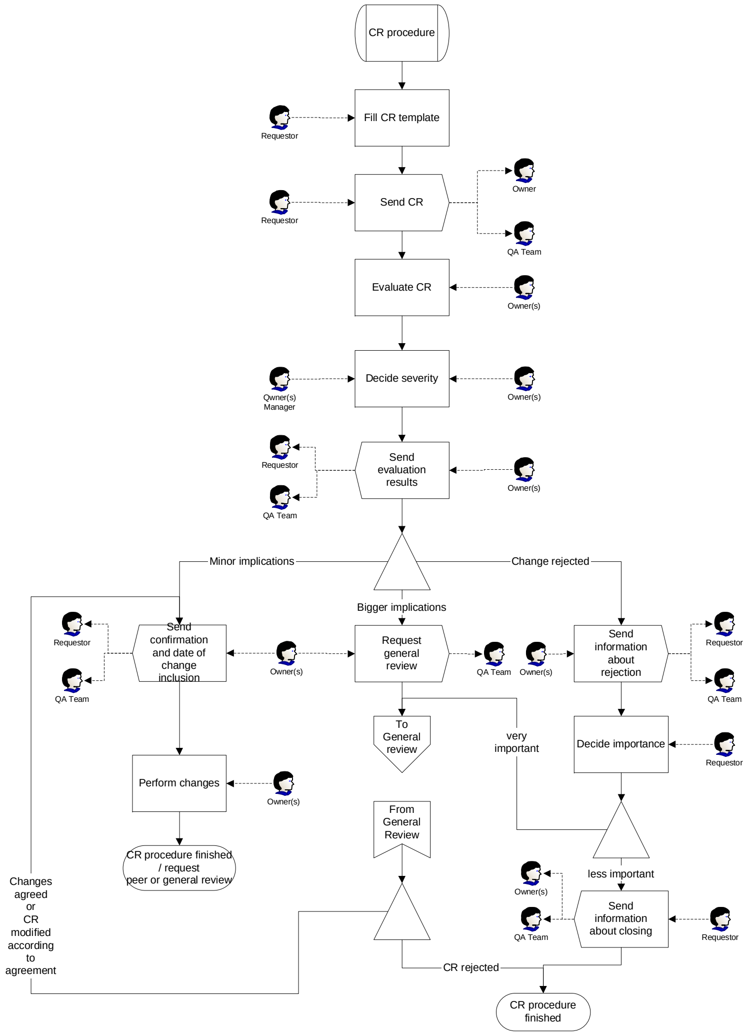


Figure 5.4.: The Change Request procedure

the implications of the change are major, a review following the general review procedure is performed. This is required to reach agreement with the leadership team and the projects executive to execute or reject the change. The third outcome of the initial change assessment is the rejection of the change request. In these cases the initiator of the change can call for a general review himself or close the change request. The process is graphically presented in figure 5.4.

5.2.3. Review procedure

The review procedure defines the roles, the essential cycles and components and the review procedures themselves required to be applied before deliverables are officially released. The purpose of the procedure is to give the Engineering Framework predefined, easy to execute means to achieve sufficient quality of product deliverables, project releases etc. It is not intend to apply this procedure to e.g. customer answer lists or other daily business write ups. It only has to be applied to real deliverables integrated into a product, officially handed over to a customer or released for internal usage. The procedure provides a framework to efficiently manage the review process and defines templates (e.g. review invitation form, review comment form, etc.) to document that all required reviews are performed correctly.

To efficiently perform deliverable reviews, the review procedure uses a two cycle approach, a peer review and a general review. The Peer review process is shown in figure 5.5. At least the second cycle, the general review, needs to be executed to transform a draft deliverable into an officially released deliverable. The most essential part of a review is/are the review meeting(s). They are necessary to achieve a final agreement in the reviewer group about the necessary changes applicable to achieve a releasable quality of a deliverable. It might be necessary to perform several review meetings. To have such a review meeting, several steps of preparation are required including a detailed check of the deliverable by the reviewers before the review meeting. The meeting itself is only intended to achieve an agreement on the necessary changes, not to perform the checking of the deliverable itself.

To ease the management of the reviews, a MS Excel workbook has been set up providing all necessary forms. This workbook contains forms like the invitations for the reviews, a review comment form to collect all comments and the review form to collect signatures. The workbook is part of the electronic project binder and needs to be filled for each deliverable or a set of deliverables. To ease the filling, the data entry is limited to changes on one worksheet propagating the contents further to all other worksheets.

5.2.4. Project management audit

The purpose of the project management audit is to assess whether a project is executed according to the development method and the other procedures. Additionally the audits intent to support the project managers in managing the projects by giving them advice. They foster to discuss the procedures and methods between the auditor and the project manager. These discussions allow to improve the methods and procedures by collecting feedback from daily operations. The audits are executed only between the project manager and the auditor. They are scheduled to be executed during the prototype development or product development stages, potentially several times depending on the lifetime of the project. They are planned during the Quality Planning process of Stage 2 to give enough time for preparations.

The audit is executed along a checklist similar to the lists in [Tav06] and [Aud06b]. The project manager has to make all project binder materials, paper

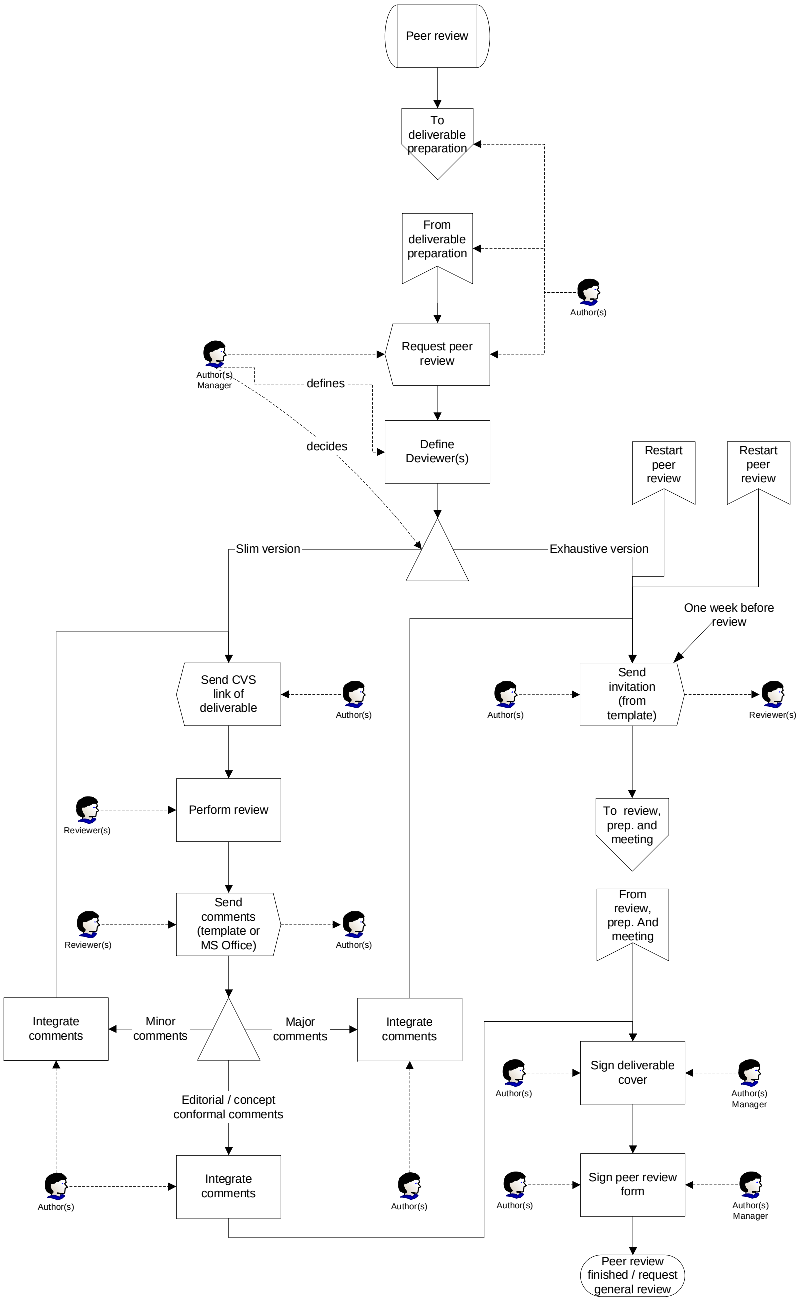


Figure 5.5.: Process for Peer Review

and electronic, available for the meeting and the auditor will selectively proof the proper usage and filling. Deficiencies and improvement potential in the management and documentation will be noted in the checklist. This checklist acts as a review protocol as well. The finalized and agreed checklist is stored as a review protocol in the electronic project binder. Further useful material for diverse audit purposes can be found on [Aud06a] and used to cover additional audit areas.

5.2.5. Project audit

The project audit performs everything a project management audit does but adds additional assessments. Examples of these additions are interviews with other project members, assessment of the current project situation (risks, issues, business case, etc.) and deliverable reviews. Therefore the projects audits can be considered to be an intermediate project gate and they are executed by two auditors of the quality department and the project team. The documentation, planning and execution of the audits is rather similar to the project management audits, only they are executed with a bigger scope. A checklist template similar to the Project Audit Templates (Prince 2 Methodology) from [Aud06a] can be used for the assessment and documentation of the findings.

5.2.6. Conclusions

The proposed quality assurance tools and components supplement the Product Engineering Method (section 3.4.3) and the Product Engineering Framework components (section 4). All these components together allow to reach the maturity level 3 of the CMMI maturity model when constantly and consequently applied. Reaching this maturity, an ISO 9001 certification is achievable with minor extensions if the organization aims for it. Via introducing measures for the different processes the organization can achieve even higher maturity levels.

6. MEMS process IP development

As has been motivated in section 3.5 the development methodology for MEMS process IP requires support for Design For Manufacturability (DFM). For the verification of manufacturing instructions for silicon based MEMS devices different methods are available. Like in other areas, the verification methods can be divided into phases. Each new phase provides more accuracy but is also more expensive in terms of time and money as shown in figure 6.1. To reduce development costs it is required to have a software environment supporting the process DFM. The approach for process DFM developed within the PROMENADE project uses a three step verification scheme. It consists of process flow consistency assessment, process flow simulation and experimental verification support. It is based on a comprehensive process knowledge management approach necessary to speed up developments and to reduce the amount of costly physical experimentation. The following sections will introduce the ideas for an environment consisting of software components supporting and enabling these three verification steps. Although the first two steps of verifying the process sequence via consistency checks and simulations are important to limit the physical experiments, the major focus of this part of the thesis is put onto the support of the experimental verification. The consistency assessment and the underlaying data representation is covered in the two PhD thesis of Andreas Wagener [Wag05] and Jens Popp [Pop05]. For experimental verification a new support and tracking concept has been invented and registered as a patent application. These concepts tie in with the system developed within the EU funded Promenade [Eur04] project. The implemented process design and tracking environment supports all three DFM phases.

The subsequent sections will describe the process DFM aspects for silicon based MEMS in more detail. While only an overview about the consistency and simulation assessment will be given in the following section, the remainder of the chapter will go into the details of the required experimental verification support. The general concepts for the capturing of the data produced by experimental verification is introduced and the realization of these concepts will be described. Additionally the evolvement of the concepts will be highlighted showing different realization approaches and their pros and cons. While these descriptions pave the ground to address reproducibility and data storing issues, the final section of this chapter describes ideas for generating new knowledge from the systematically captured data.

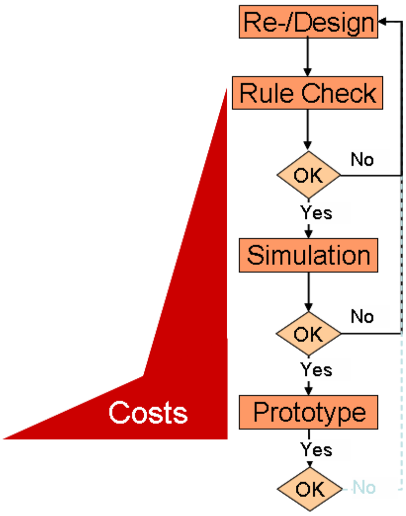


Figure 6.1.: Three step verification flow and costs

6.1. Process DFM components

As introduced in [WPS+05], [WSP+06], [SOH+06] and depicted in figure 6.1 process DFM can be considered a three step approach. It needs to use process flow consistency assessment, process flow simulation and experimental verification to verify a process design. Section 2.3 outlines the surrounding and the necessities for MEMS process IP developments. Figure 2.2 presents the vision for a process design and tracking environment addressing all three DFM areas. A more detailed presentation of this environment vision is given figure 6.2 showing the interactions between the different environment components and the outside world. The three areas of verification are briefly described in the following subsections.

6.1.1. Verification of Manufacturability

The first and least expensive step in the verification of MEMS manufacturing processes is a formal verification based on the process flow model, the consistency check. In this phase the process flow is checked using rules applied to the process steps of the flow. These rules are abstract process knowledge stored in the system by the process engineers. Each rule can demand or recommend the compliance with certain criteria while a single criterion can be the pre- or post-processing of the process step or process step group. Additionally it is possible to forbid (NOT operation) or recommend against the usage of certain steps or parameter values. Some rules are only valid if certain conditions are met. Conditions could be the usage of certain process steps or groups or the matching of certain parameter values. They can be used to activate, respectively deac-

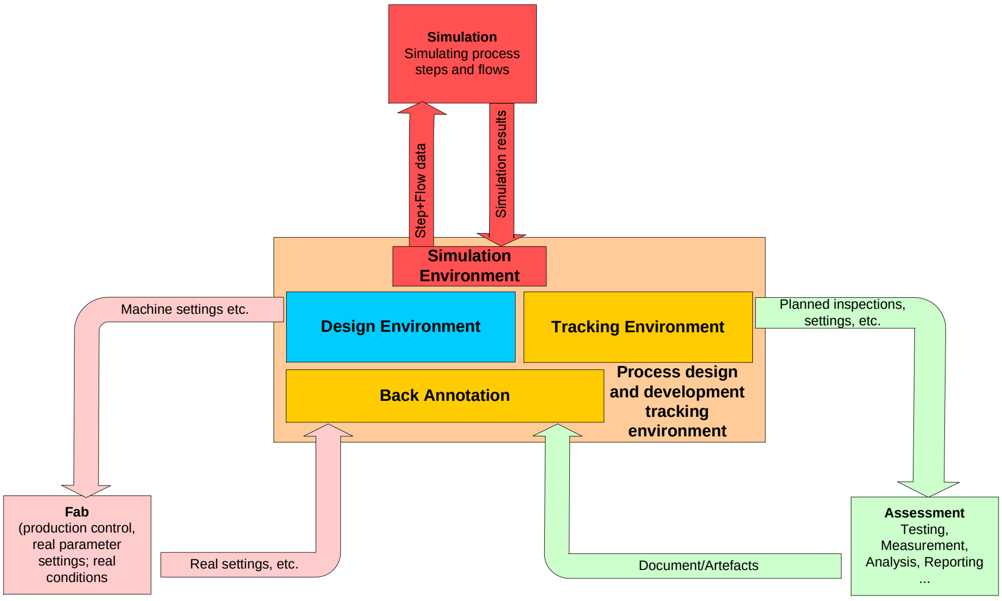

Simulation
Simulating process steps and flows
Step+Flow data
Simulation results
Simulation Environment
Machine settings etc.
Design Environment
Tracking Environment
Planned inspections, settings, etc.
Back Annotation
Process design and development tracking environment
Fab
(production control, real parameter settings; real conditions
Real settings, etc.
Document/Artefacts
Assessment
Testing, Measurement, Analysis, Reporting
...

Figure 6.2.: Decomposed environment showing the major blocks of the process development and tracking system

tivate, rules. Finally it is possible that a process step must comply with more than one rule or that only one of a set of rules must be met. These scenarios can be supported by connecting the rules using the Boolean operations AND and OR. To enable the effective use of such rules, it is necessary to define a data structure that not only allows a fast access but also allows the inheritance of rules. The inheritance enables the user to define rules for groups of process steps. With the usage of inheritance the user only needs to assign the rule to process step groups - all members of the group inherit the rule. More details of this verification approach are published in [Pop05].

6.1.2. Verification by Simulation

The verification of manufacturability indicates only the feasibility of the process flow and does not assess the physical results of the processing. To ensure that a process flow leads to the desired results, it is necessary to verify the flow either by simulation or experiment. Though experimental verification gives the most reliable results, it is also very expensive and time-consuming. Especially in early design stages, when the process designer still experiments with different process steps, parameters, and lithography masks, a fast feedback is of higher importance than physically exact results. During these stages process simulation is the most appropriate method for verification as it is comparatively fast and inexpensive. For a first assessment of the applicability of a process flow a simple cross-section with rectangular structures is usually sufficient. Later on when the design converges, more and more details gain importance, so more sophisticated and specialized simulation models are required. The PROMENADE system is designed to cope with these requirements: The process steps are arranged in a hierarchy with the most general types of process steps at the top. To each of these types a very basic simulation model is assigned. The simulation model is passed on through the hierarchy, thus ensuring the availability of at least one simulation model for every process step. At any point in the hierarchy a more specialized model can be added, so that each simulation can be performed with the necessary level of detail. To perform process simulations of a certain process flow, the PROMENADE environment has to look up the simulation models for each process step of the flow in the knowledge base. If more than one appropriate model exists, the designer has to select which one to use. When models for all process steps of the sequence are selected, the environment automatically parameterises them with process and material parameters from the knowledge base. After that it starts the different simulation tools that actually perform the simulation. Finally the results are presented to the designer who has to decide whether the process meets the requirements. If satisfactory results are achieved in the process design system, the experimental verification can be started supported by the experiment tracking system.

6.1.3. Verification by Experiment

After the first two rule- and model-based verifications, the Experimental Verification proofs the manufacturability in a real fabrication line. Because this step implies the production of real silicon, the verification by experiment is the most expensive one. This is due to the involved production costs themselves as well as the time these experiments require. For silicon based MEMS designs the flow of the experimental verifications involves typically at least the following steps: (1) Setting up an experiment plan, potentially supported by a Design Of Experiment (DOE) system. (2) Defining different lots, their purposes, analysis and tests to apply and the amount of wafers per lot. (3) Processing of lots and their wafers where different wafers might receive slightly modified processing. (4) Performing of different analysis during the processing sequence. (5) Analysing and testing the wafers/devices. (6) Performing failure analysis of the wafers and devices. (7) Comparing experimental results with design expectations.

This list shows that relations between experiments, lots, wafers, process steps, and process flows (all referred to as development dimensions) in relation to the different assessments and their results exists. These relations are graphically presented in figure 6.3. The assessment results are often stored in files of various formats (referenced as documents or artefacts). They are related

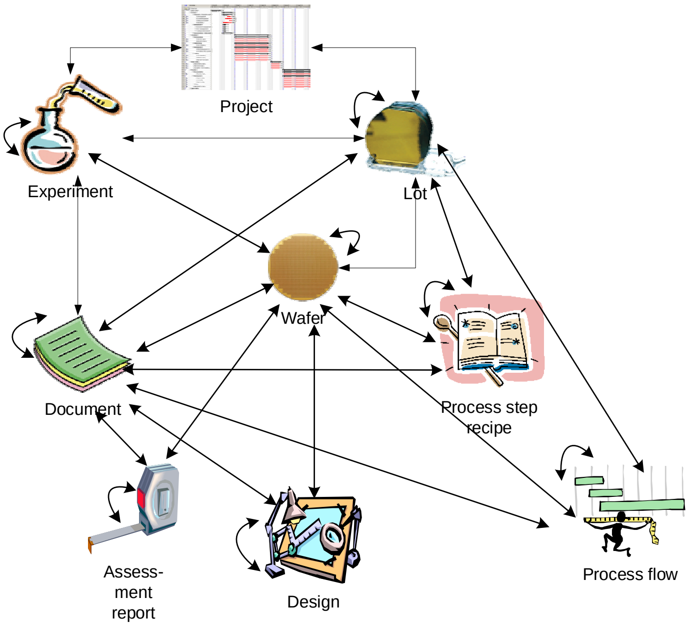


Figure 6.3.: Development dimensions and relations(relation types reduced to a minimum)

to the development project, to the development dimensions and to the process step and flow recipe. The designs used to manufacture the wafers are a special type of file with explicit visioning, displayed as the Design dimension. Results of tests and assessments are summarised in the special dimension Assessment report referring to the test result data and to the corresponding wafers the test were performed on. The experimental verification efforts generate MEMS processing knowledge, which needs to be collected and used to recalibrate the initial process design. As can be seen from the list, the real verification step is the comparison between the real results obtained by the analysis and the results expected after the design phase. The other steps are basically preparations for the verification similar to the process rule collection for the Verification of Manufacturability described in section 6.1.1. For the real assessment of the analysis results and the comparison with the design expectations, several different analysis tools (e.g. Six Sigma analysis, Statistical Process Control) can be used. To make these verifications successful, the required input data for these tools needs to be captured carefully and systematically by a development tracking system. The necessary support tools are described in more detail in section 6.3.

6.2. Overall approach

The objectives of the PROMENADE project are to realise a computer-based environment supporting process engineers in creating, verifying, simulating, optimising and maintaining thin film Si processes with predictable characteristics. Additionally it is supposed to support designers of microdevices by offering them a formal interface to constraints from the technological domain and facilitating design for manufacturing. Special attention has to be paid to respect IPR protection issues of users and the different partners and company specific IPR protection.

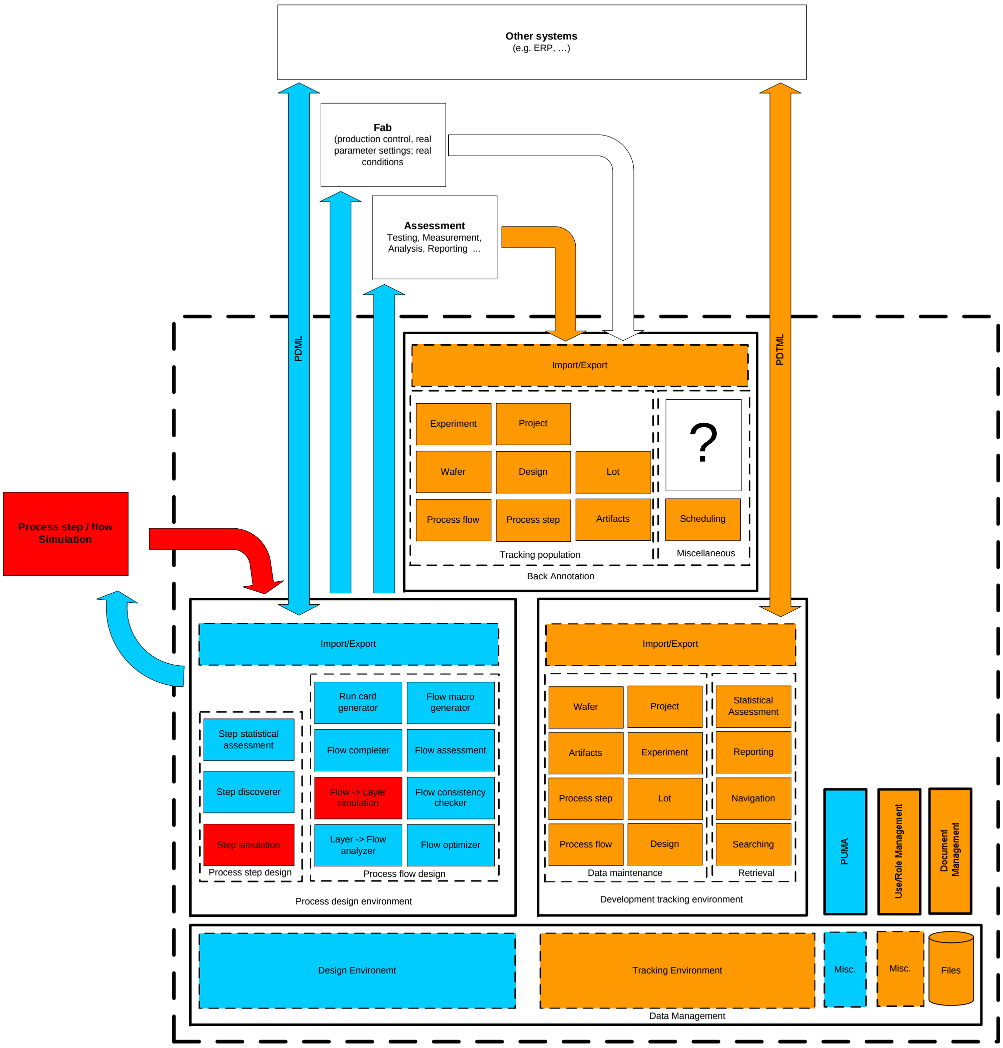


Figure 6.4.: Promenade system decomposition

A rough overview of the envisioned environment and its outside world interactions is given in section 2.3 and in figure 2.2. This figure has been detailed further in figure 6.2 to show the different main components inside the PROMENADE environment. Detailing this system vision further down, having the overall experimental verification sequence in mind, results in the Promenade system decomposition presented in figure 6.4. The drawing presents that the overall system consists of the seven major components Process Design Environment, Development Tracking Environment, Process Step + Flow simulation, Back Annotation and the shared components Document Management, Parameter and Unit Management (PUMA) and the User/Role Management.

Focussing on the Design for Manufacturability aspects, the above described area of Verification of Manufacturability is covered by the modules "Flow consistency checker" and "Flow assessment" of the Process Design Environment. The "Verification by Simulation" aspects are covered by the red marked modules of process step simulation, flow→layer simulation and the whole module Process step/flow simulation. The aspects of support for Experimental Verification is covered by the Development Tracking Environment and will be described in more detail below.

While the blue and red parts of figure 6.4 are realized by other partners of the Promenade consortium, the orange parts are realized by the efforts for this thesis. The question mark in the component Back Annotation is a long lasting research activity for (semi-)automatic knowledge generation out of the data of the Tracking Environment. For this question mark area only idea generation and first concept work has been performed. These activities are described in section 6.6 about the Back Annotation and chapter 7 Further work.

6.3. Experimental verification support

To make the assessments of the experimental verification comprehensive and successful the data generated during the efforts needs to be captured carefully and systematically. This verification data consists of analysis results, process step and flow data, data about the verification progression, discussions and so forth. This data perspective is described in more detail in [SOH+06]. Currently there is no structural methodology for a systematic collection of the verification data limiting the assessments, the tool usage and the knowledge gains. As published in [OVC05] and outlined in section 2.2.6, the data collection is typically done ad hoc with results being kept on paper, in spreadsheets, on some file server or just in the minds of the process and test engineers. This implies that the data is in general not easily accessible for any process changes, future process developments or knowledge retrieval out of the data. Another disadvantage of this non systematic data collection is that the initial development efforts are not reproducible and the data is not available for review by other developers.

Therefore the development of a structural approach for capturing and evaluating the experimental verification data is essential. The following sections present the analysis results together with the developed structural approach. This approach has been implemented into two software systems described after that.

6.3.1. Analysis results

Analyzing the process of experimental verification, as it is described in the State-of-the-Art section 2.2.6, and the items used for that purpose, the following coherences can be detected. The process design verification task belongs to one development project and uses a set of experiments to perform the necessary evaluations. To perform those experiments a number of lots is used and run through the fabrication line. Each lot consist of a number of silicon wafers. Further the used process flow and its variations for the different experiments is assembled from a set of process steps. Therefore a process flow can be decomposed into an ordered collection of process steps as listed on a fab runcard. These decomposition structures of the process flows and the verification entities (experiments, lots, etc.), is presented in figure 6.5. The wafers of one lot potentially receive slightly different processing using different charges. This implies that the different wafers of one lot might relate to a different process recipe because steps in the flow might be exchanged or modified. Additionally different runs of the same flow or step recipe will potentially lead to slightly different result properties because of inherent process step variations. This causes differences in verification data collected during or after the execution of the processing. Often results of assessments are stored in files of various formats (e.g. SEM-image file, XPS analysis text file, etc.; referenced as documents or artefacts). They are related to the development project, to the development dimensions entities and to the process steps and recipes. The wafers are manufactured using a specific mask design which are special documents because they require explicit version management. This implies that it is required to be able to relate each wafer with a certain design. The test and characterization work performed on the wafers and the raw test result data need to be captured in Assessment Reports and related to the wafers. Taking these potential relations into account it can be said that possibly every entity used or produced during experimental verification can be related to every other. This is due to the fact that similarities between items, similar results or other identified correlations can and should be expressed as a relation. Such a scenario of relations is presented in figure 6.3 and 6.5 where the relations between the different dimensions are reduced to the most common and important ones.

Further it can be noted that files containing the experimental result data are often stored in an unstructured way or in a highly specific structure on a file server. In the case of the unstructured storage, knowledge about the relations,

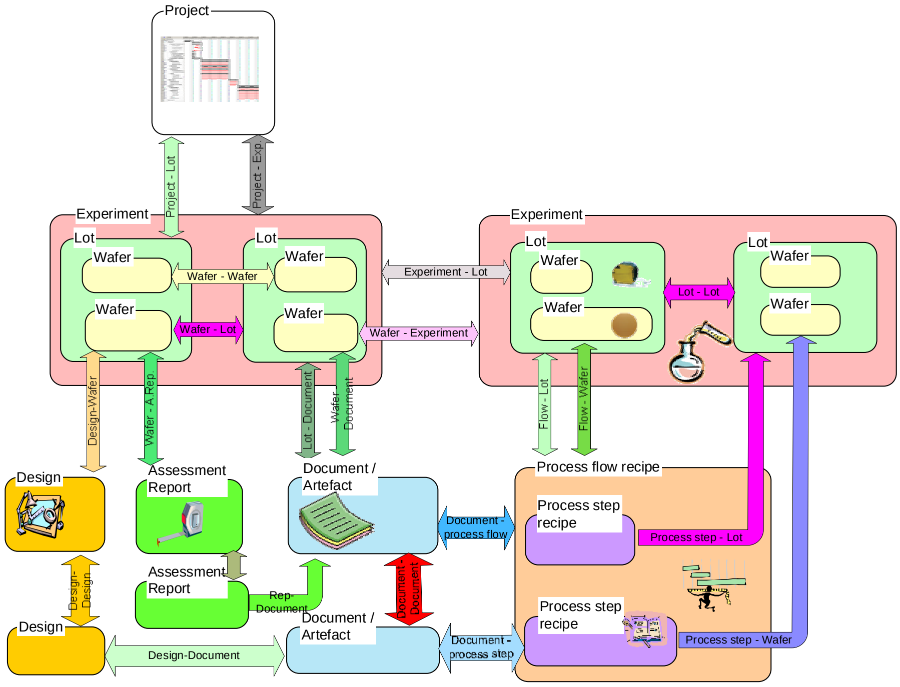


Figure 6.5.: Decomposition of verification entities(relation types reduced to a minimum)

similarities etc. is lost. Using a highly specific structure allows only to capture one kind of similarity or relationship and disregards the others causing knowledge loss as well. One major reason for this dilemma is the lack of a generic categorization to structure the data capturing all kinds of similarities and information. But the information about the similarities is essential for a thorough assessment and evaluation of the results. Therefore it is necessary to capture the information to be able to access the data from more than one point of view to catch every possible aspect. To allow this a more general and flexible way to store the data than just files in a directory is required. Similar findings are reported in [Bro98]. A novel approach addressing this is described in in following section.

6.3.2. Structural approach

As discussed in previous sections, today technology information derived from simulation, tests or measurements is usually kept informally and non-systematically on paper, in Excel sheets or merely in the minds of process engineers. Therefore it is hardly accessible for use in future process development projects. The concept for experiment verification support and tracking systems allows the tracking of the designed process steps, flows and verification entities running through a fabrication line together with their result artefacts and assessment reports. It is based on the idea of building a system for managing and tracking of an arbitrary number of information entities types (e.g. process steps, process recipes, experiments, lots, wafers, artefacts). Each entity can be cross-referenced to any other entity, of the same type or a different type. Thereby a network of information is created, in the mathematical and computer science terms an undirected graph with different vertex types. A graphical representation of such an example information network is presented in figure 6.6. It is presented in analogy to a spider web having the different development dimensions (entity types) as axes. The relations between the items of the different dimensions form connections between the axes. Relations between items of the same dimension can be considered as running in parallel to the axes. A further analogy to a spider web or a graph can be detected in saying that the direct relations between items are extended with less strong relations (less similarities) if items are related via other items. This way the information network can be navigated while using the relations to analyze similarities and correlations. The more hops between two items, the less they are similar or correlated. It is intended that the information network can be accessed by several software modules operating on the network. They should supply functionalities for information maintenance, reporting, analysis, visualization and searching. This information representation can be used to build software systems representing and modelling the data of the experimental verification as described below.

A key feature of the approach is the document and artefact management for the different documents/files. The document management offers the ability to build a wrapper around file based verification results and all other kinds of files to attach meta-data to it. Additionally it allows the tracking and linking of this data to the other verification data entities. The meta-data that can be attached to the documents and artefacts can include keywords, descriptions, names or any other information that could be useful to associate with the files.

Each entity type has it own information maintenance module allowing the entry, modification and removal of entity attribute data. The entity types contain all required meta-data ranging from fields like descriptions, names, purposes, etc. to the progress in the fabrication line (e.g. movements in the line, status change, priority change, reaching certain milestone). The history of each tracked entity is kept together with the date and time these changes occurred and optionally comments of the operators or from foreign systems. This enables the easy reporting of the lot, experiment and project status and the creation of reference knowledge for future developments and verification planning. Based on the historic data new developments and experimental verification efforts can be planned more efficiently by using the historical data as planning basis.

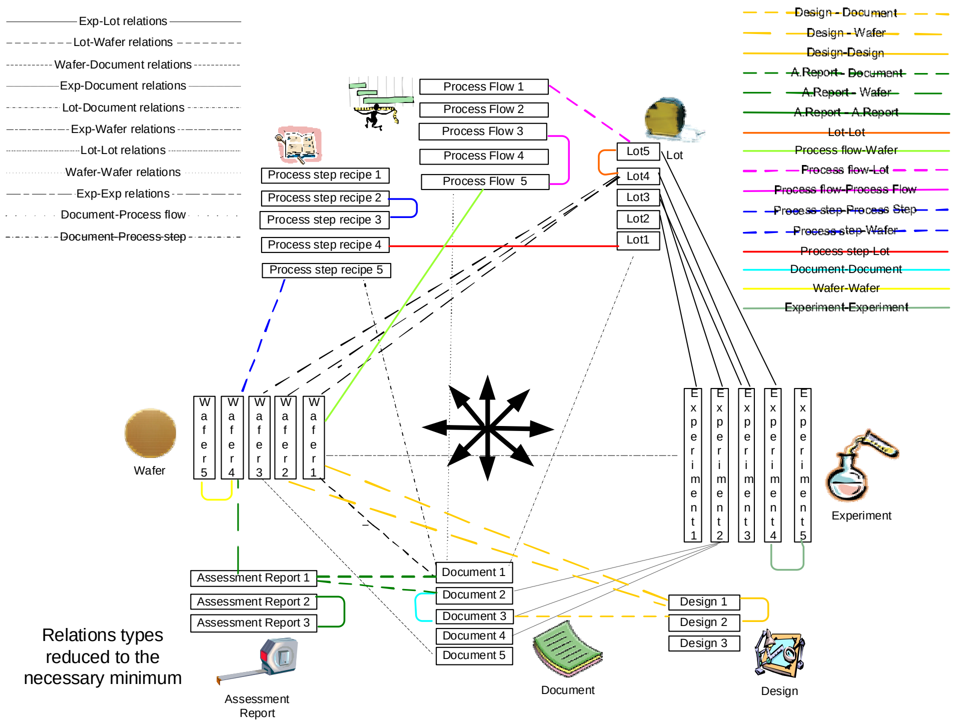


Figure 6.6.: Information network or graph with data dimensions

All these features together allow a structural approach to track and manage experiment verification data. In fact the basic ideas behind the approach are generic for several industries. They can not only be used for silicon based MEMS developments but could be used for the development and verification of medical substances, etc. The approach is applicable for all developments having a decomposition structure, producing verification data in same kind of file and where the relations between the files and the verification entities build a major part of the verification knowledge. Therefore the approach has been published as a patent application in [Ort05b].

Up to now the systematic approach has been implemented two times into software systems for the support of silicon based MEMS process development. Both systems have slightly different purposes and use a different technical foundation. Both implementations will be described in the following sections but the first implementation in less detail.

6.4. Process Development Tracking System(PDTS)

The Process Development Tracking System (PDTS) is the first implementation of the systematic approach described above. The following descriptions will briefly described the history and development of the system to motivate the solution approach. Further more some technical details and a brief overview of the system will be presented.

6.4.1. The History

The PDTS implementation can be considered a requirements evaluation prototype. Because standard requirements engineering techniques were difficult to apply to this field, a prototyping approach was selected to evaluate the requirements. The main problem for the inapplicability of requirements engineering technique is the limited awareness of reproducibility issues by the process engineers and their attitude of ”Being happy with what we have”. This pattern was observed in several examples inside the PROMENADE consortium. Because of that the chosen approach was to get an overview about the work procedures and the items used to perform the experimental verification. This knowledge together with results from diverse interviews with the process engineers has been used to develop the structural approach and the initial system concept. The initial implementation has been set productive and the engineers slowly started using the system. While using the system, misinterpretations of requirements and work procedures, new requirements etc. were detected, discussed and the system was changed or extended. An example for newer requirements not implemented into the PDTS prototype anymore are the explicit modelling of designs and the support for capturing the assessment reports. The efforts on the PDTS prototype laid the baseline for the second implementation of the approach inside the PROMENADE project because the requirements and concepts were already evaluated and worked out. The PDTS implementation of the structural approach has been published in [OVC05].

6.4.2. System architecture

The PDTS system uses a client-server based architecture with two/three tiers as presented in figure 6.7. A more detailed description of the different architecture possibilities is given in section 6.5.4. The user interface uses a web based client running in a web browser. The middle tier or application layer is realized with a web server using dynamic web pages. The persistence layer uses a relational database management system in the background. File based experiment verification data is kept as files on some server and is only referenced. For referencing any kind of URL (file, http), as indicated in figure 6.7, is used. The referencing has the drawback that renaming, moving or deleting of the source surrenders the references useless. To limit the amount of ”broken links” the system contains means to check and change patterns in the URLs to keep the references intact. For simple file references a favourable solution would be to use a regular document management or revision control system as a foun-

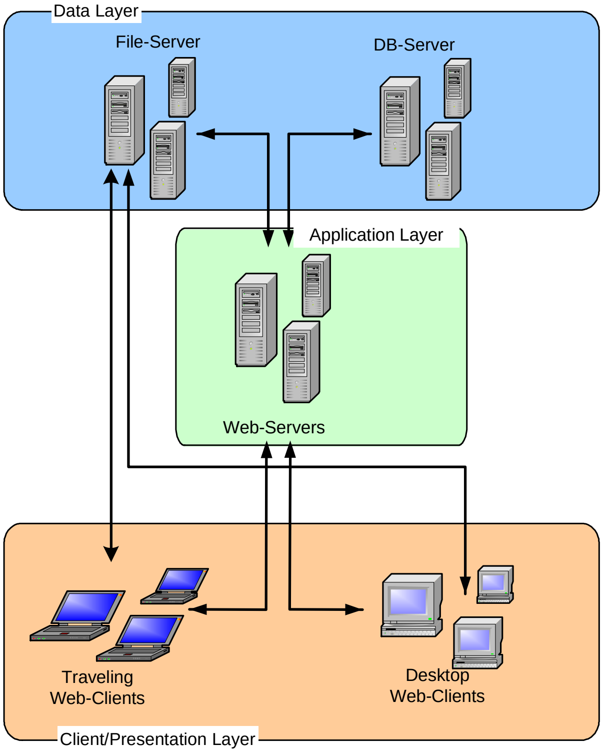


Figure 6.7.: PDTS system architecture

dation keeping track of the documents. The implementation of this has been postponed to the newer system described below.

One of the requirements for this prototype has been to provide a slim interface for thin clients and possibly no direct client application at all. Therefore and for rapid prototyping purposes it has been decided to implement the system as a web-application only using the web-browser on the client-side. For the generation of the dynamic web pages PHP has been selected because it supports sufficiently the rapid prototyping. As a Database Management System the Oracle database has been chosen because the developed data-model uses foreign-key constraints, the system should run on Linux and the system should be scalable.

6.4.3. The System

Although the PDTS system has been developed as a prototype it is in productive use for more than three years. It is still used to evaluate new ideas and to propose solutions for detected problems. This is due to the rapid prototyping technical foundation making it easier and more efficient to implement new features into this environment first. Additionally the system is used for remote development sites as a means to transfer and control development efforts. Migration means from PDTS into the new EVERIST system have been implemented to facilitate the data reuse and collection from different sources.

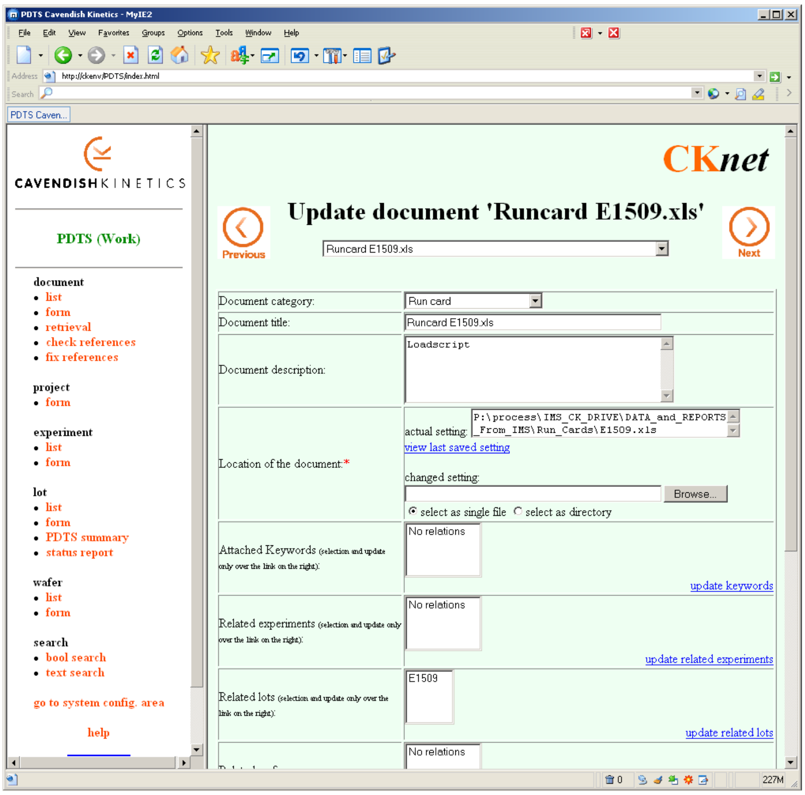
Information management

The PDTS system realizes the approach described in section 6.3.2 with the limitations of only supporting the entities projects, experiments, lots, wafers and documents/artefacts. It assists the process engineer in tracking current development efforts and experimental verification data as well as keeping a structured, linked history of past development efforts. The system is able to store and manage all development-related data which is linked to different items on different levels of the development hierarchy. These results may be correlated with each other. Additional to those, the system is extendable with additional meta-data fields and is able to be extended to handle additional development data dimensions like die and single chip data or the dimensions described above. The systematic collection of data also allows for comparing the real test data with the predicted result, thus allowing rapid, intelligent iterations.

Of special importance are the predefined meta-data fields like status, milestone and priority and the time save histories of changes of these properties. PDTS is capable of tracking every status change for all experiments, lots and wafers. Even the status after each applied process step is storable, potentially with comments from the operators. Alternatively information from a Manufacturing Execution System (MES) could serve as input for the system, thus enabling simple report generation and automatic recipe tracking. The histories allow further knowledge gains because they allow reporting on progressing times and therefore allow better development planning based on historically acquired data.

Document management

A key feature of the PDTS system is the document management, or the ability to build a wrapper, around ”non-standard format” process data. This allows the tracking and linking of this data as well as the attachment of meta-data to it. The attachable meta-data can include e.g. document categorization, descriptions, keywords or any other information that would be useful to associate with the process information. An example for the document meta-data is presented in figure 6.8. Additionally, the system can link the documents together with the different levels of experiments, lots and wafers, also shown in figure 6.8. This is required in order to see the impact of any recipe change on the results as well as getting a better understanding of the cause-and-effect-chain of these changes. The relating and referencing is achieved based on a loose collection of documents referenced by URLs (file, http) as described in the System Architecture section 6.4.2.

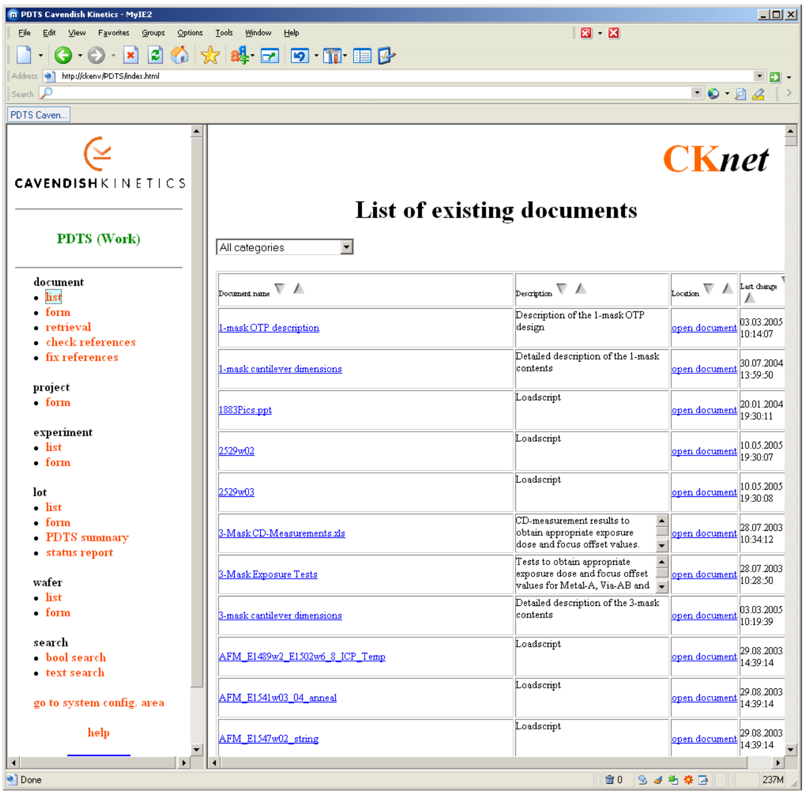


Update document 'Runcard E1509.xls'

Runcard E1509.xls

Document category:	Run card
Document title:	Runcard E1509.xls
Document description:	Loadscript
Location of the document:*	actual setting: P:\process\IMS_CK_DRIVE\DATA_and_REPORTS_From_IMS\Run_Cards\E1509.xls view last saved setting changed setting: Browse... ● select as single file ○ select as directory
Attached Keywords (selection and update only over the link on the right):	No relations update keywords
Related experiments (selection and update only over the link on the right):	No relations update related experiments
Related lots (selection and update only over the link on the right):	E1509 update related lots
	No relations

Figure 6.8.: Meta-data form for documents/artefacts



PDTS Cavendish Kinetics - MyIE2
File Edit View Favorites Groups Options Tools Window Help
Address http://ckenv/PDTS/index.html
Search
PDTS Caven...
CAVENDISHKINETICS
CKnet

PDTS (Work)

document
- list
- form
- retrieval
- check references
- fix references

project
- form

experiment
- list
- form

lot
- list
- form
- PDTS summary
- status report

wafer
- list
- form

search
- bool search
- text search

go to system config. area

help

List of existing documents

All categories

Document name	Description	Location	Last change
1-mask OTP description	Description of the 1-mask OTP design	open document	03.03.2005 10:14:07
1-mask cantilever dimensions	Detailed description of the 1-mask contents	open document	30.07.2004 13:59:50
1883Pics.ppt	Loadscript	open document	20.01.2004 19:30:11
2529w02	Loadscript	open document	10.05.2005 19:30:07
2529w03	Loadscript	open document	10.05.2005 19:30:08
3-Mask CD-Measurements.xls	CD-measurement results to obtain appropriate exposure dose and focus offset values.	open document	28.07.2003 10:34:12
3-Mask Exposure Tests	Tests to obtain appropriate exposure dose and focus offset values for Metal-A, Via-AB and	open document	28.07.2003 10:28:50
3-mask cantilever dimensions	Detailed description of the 3-mask contents	open document	03.03.2005 10:19:39
AFM_E1489w2_E1502w6_8_ICP_Temp	Loadscript	open document	29.08.2003 14:39:14
AFM_E1541w03_04_anneal	Loadscript	open document	29.08.2003 14:39:14
AFM_E1547w02_string	Loadscript	open document	29.08.2003 14:39:14

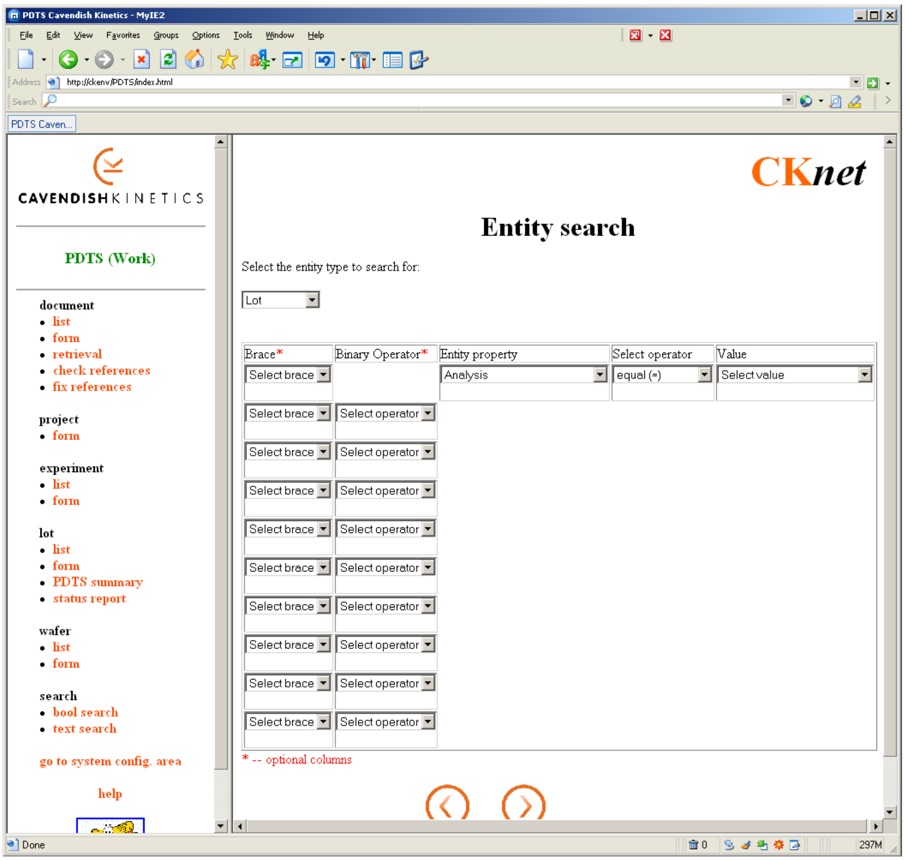
Figure 6.9.: List of documents in the system

Filling the system

To fill the system with as less effort as possible, PDTS contains features to automatically populate the database from diverse sources. This includes the population of the document storage from the file system. Additionally relations between corresponding entities can be created as well as keywords and comments can be attached to the created entities. On top of that, retrieval and population of the data network from e.g. Excel sheets is supported. The population uses the special user and description ”Loadscript” as indicated in figure 6.9. All automatically populated entities are filled with ”Loadscript” in certain fields to indicate that they have been created by the automatic loading and that the fields might require update and filling.

Retrieval

Besides the archiving of the development data, the retrieval is of major importance. For that navigation functionalities are implemented providing for navigating the data network. The development hierarchy can be easily and directly retrieved via a drill down functionality. Additionally diverse list presentations e.g. lists like in figure 6.9, allow the navigation through the information net-



PDTS (Work)

document
- list
- form
- retrieval
- check references
- fix references

project
- form

experiment
- list
- form

lot
- list
- form
- PDTS summary
- status report

wafer
- list
- form

search
- bool search
- text search

go to system config. area

help

Entity search

Select the entity type to search for:

Lot

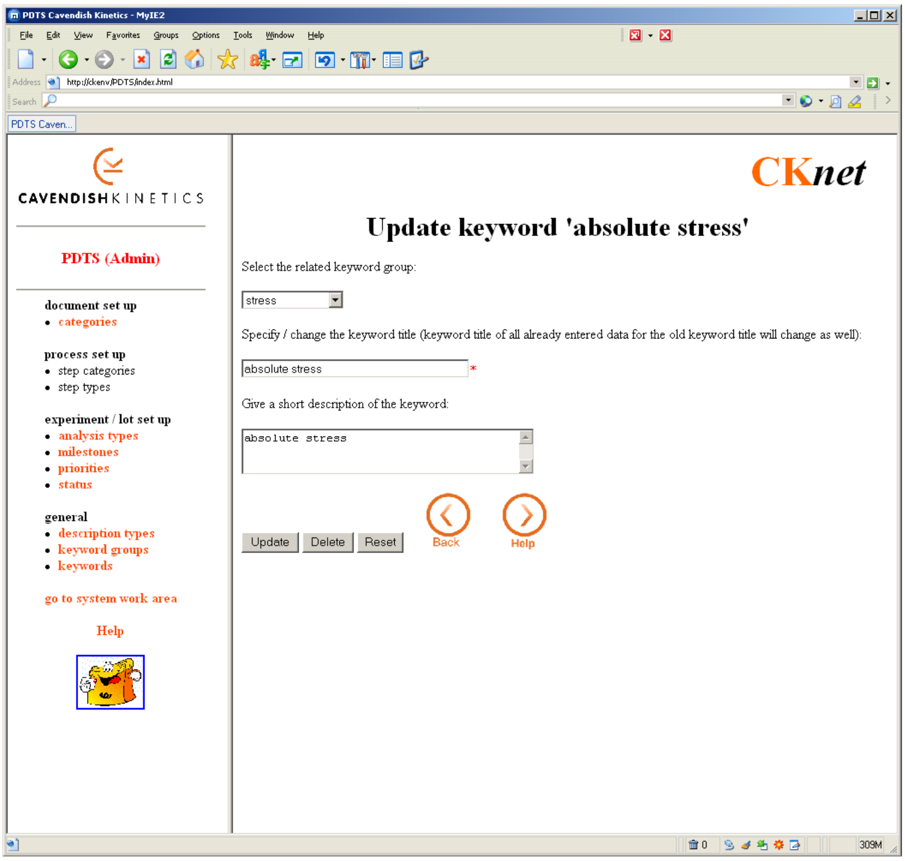
Brace*	Binary Operator*	Entity property	Select operator	Value
Select brace		Analysis	equal (=)	Select value
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			
Select brace	Select operator			

* -- optional columns

Figure 6.10.: Entry form for defining a search function

work. The drill down functionality allows to navigate from the form of one entity directly to a related entity by clicking on the names of related entity, as shown in figure 6.8. From there the further related entities can be accessed. Additionally the list of entities of the same type can be iterated via the previous and next buttons shown in the figure.

For further retrieval possibilities, the PDTS prototype offers flexible search mechanisms powered by a search engine enabling specifically tailored searches of the database contents. The main search mechanism consists of a generic engine mapping all entity attributes to the database fields of all tables. For that a table driven Boolean algebra is created and the user can define any Boolean function over this algebra. The entry screen for the search function is presented in figure 6.10. The search engine converts the defined function into the corresponding SQL queries and retrieves the matching entities form the database and presents them as a list. The implemented engine is completely generic and reads its configuration from a set of database tables. If attributes are added to the entities the engine remains the same, only some additional database entries in the search configuration tables need to be added. With that



Update keyword 'absolute stress'

Select the related keyword group:

stress

Specify / change the keyword title (keyword title of all already entered data for the old keyword title will change as well):

absolute stress *

Give a short description of the keyword:

absolute stress

Figure 6.11.: Administrator menu and the keyword configuration areas

the search mask is amended automatically and the user is able to search for the new field.

An additional search mechanism is the full text searching over all text fields of the database. This search mechanism is build on top of the Boolean search, just defining the appropriate Boolean function to search all text fields for the requested search string.

Administration area

The PDTS system is split into two different areas, the user area and the administrator area. Both have different different menus. The menu shown on the left side of figure 6.9 represents the user menu. The administrator menu is shown in figure 6.11. The menu gives the selection to manage all configurable and selectable attributes used by the entity types. These attributes consist of e.g. document categories to sort the different documents, analysis types, status types and keywords. Additionally the figure shows an example form to manage a single keyword and its grouping into a keyword category.

Feature summary

- Interface to the system via a web browser using dynamic web pages
- Relate, track and search not only experiments, lots and wafers, but also inspection reports, measurements, test results and any other documents or files generated
- Easy archive feature to enable straightforward review of past developments
- Tracks all results and makes them easily retrievable and reportable
- Automatically populates database with information from the file system and files
- Easy tracking in time of experiments, wafers and lots or any other development dimension
- Powerful, easy to configure search feature for all types of data
- Flexible attachable texts, grouped into types and attachable to every item type. Additionally the texts can be related to each other. This can be used to sort all conclusions of an item to their related purposes. An overview list shows the matching pairs to get an easy to view comparison between purposes and conclusions.

6.4.4. Data model

The data model has been developed with the picture in mind to treat all item types as independent entities and to enable the relation from every item to every item. All item types are mapped to different database tables, although their attributes are relatively similar. This was done to be able to easily extend the data model and to partition the data as much as possible.

The overall data model of the PDTS system is presented in figure 6.12. Because this is too complex to follow figure 6.13 shows only the part relevant for the experiments in a condensed drawing. The models for the other item types are very similar.

The relations between the item types are kept in separate tables per type pair (e.g ExpLotRel in figure 6.13). Therefore every type combination e.g. relation between experiment and lot has its own table to be able to use foreign keys to directly keep data consistency. Because the relations are considered as undirectional only one table per pair exists, e.g. there is a table experiment to lot but no table lot to experiment. Therefore the systems logic always has to take care about the symmetrical treatment of the relations.

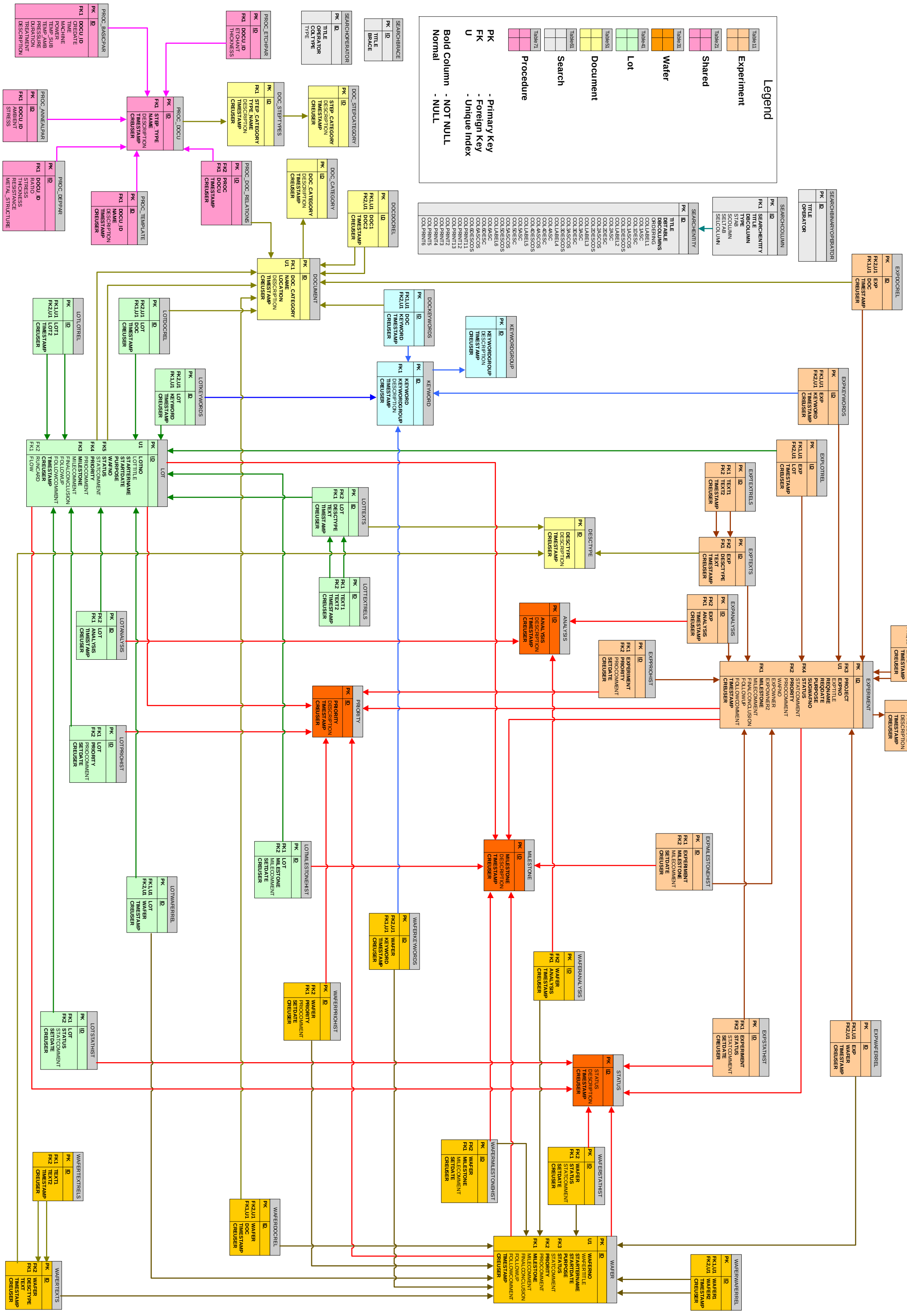


Figure 6.12.: PDTS complete data model

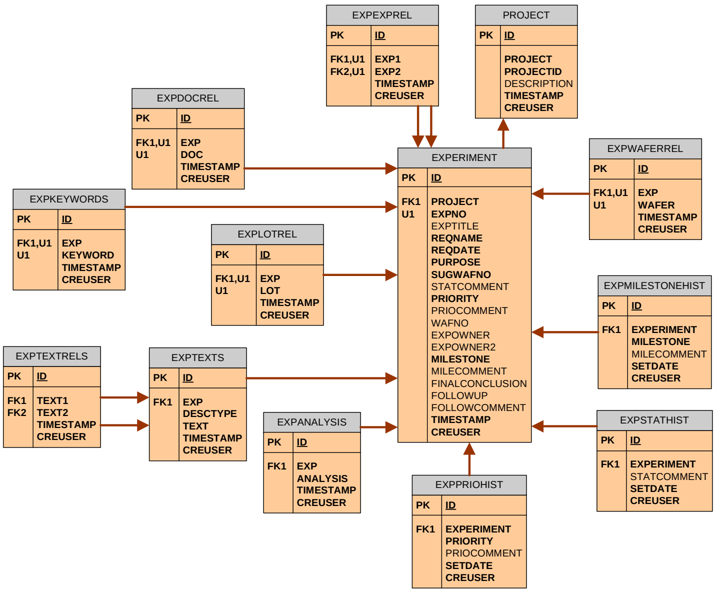


Figure 6.13.: PDTS experiment data model

Each selectable attribute, like milestone, priority etc., was build into a separate object mapped to one separate table as well. These basic configurations like analysis types, milestone types, status types, priority types and keywords are maintained in the separate administration area of the system (see figure 6.11). In the forms of the different entity types they can only be selected from a list to attach them to an item. For some on these attributes (e.g. milestone, status, priority) even a history, together with a operator comment, is kept to see all changes during the development. These changes are kept in separate tables per item type per attribute implying the existence of an experiment milestone history table, experiment status history table etc. (compare ExpMilestoneHist, ExpStatusHist, etc. in 6.13). This is due to performance reasons and to allow a very tight data modelling providing for strict consistency checking by the DBMS using foreign key constraints.

The texts attachable to every item type are separated out into a table per item type as well (compare ExpTexts in 6.13). The spilt out is done to be able to attach as many texts as possible to an item and still keep the consistency. All these texts contain a free text attribute together with a text type e.g. purpose, conclusion where the text types are configurable in the administration area. An additional table per text item type holds references between individual texts (compare ExpTextRel in 6.13). They enable to relate texts belonging together e.g. the purpose of an experiment and the resulting conclusions addressing that purpose.

6.5. Promenade Tracking Environment (EVERIST)

The second implementation of the structural approach (section 6.3.2) was done within the PROMENADE project as its Tracking Environment (EVERIST system). It is in principle a re-implementation of the concepts and features previously implemented into the PDTS systems. But additional ideas and features have been integrated into the concept and have been implemented afterwards. Because of that the descriptions in this section will be limited to the differences and additions of the re-implementation. The further descriptions will first give a very brief overview about the overall system. Then the new concepts and added and changed features will be described in detail. Afterwards the systems architecture options and the finally chosen and realized architecture will be described. This is followed by the data model and the embedding of the Tracking environment into the overall PROMENADE system.

6.5.1. The System overview

The current section briefly gives an overview about the overall EVERIST system. This is to give the reader a better orientation for the further descriptions. Here only the very basic and re-implemented functionalities will be described.

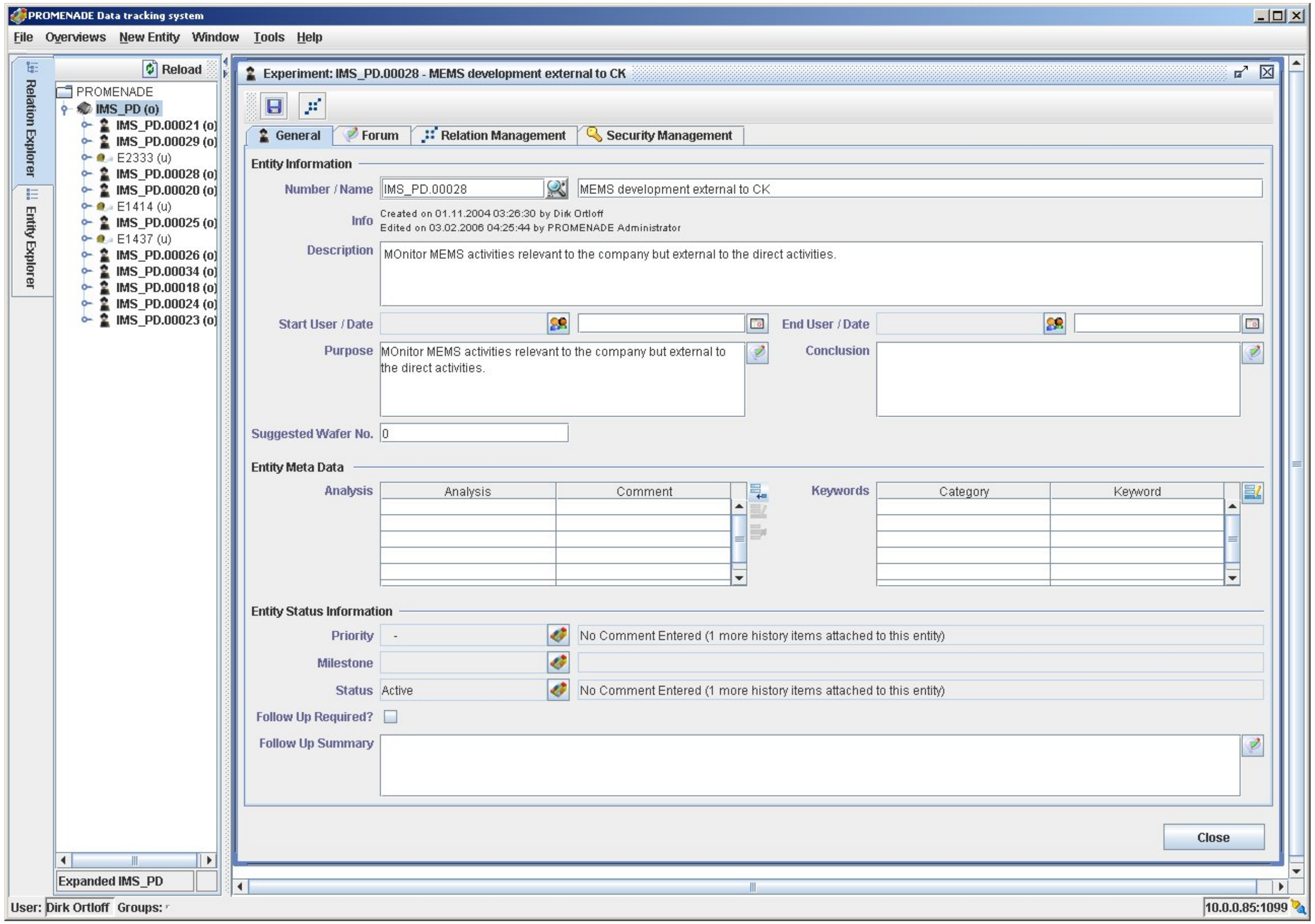
The EVERIST main window is presented in figure 6.14. The window is separated into three areas. The top area presents the menu and the button panel. The left panel contains the two main navigation possibilities Relation Explorer and Entity Explorer (both described below) allowing the user to navigate through the objects. Via the ”Open” option of the context menu of these explorers an entity can be opened and via opening the folder of the entity the related items are shown. The right main part of the window presents the entity data as in the figure for the general tab of an experiment.

Data Maintenance

The general forms for the different tracked entity modules (entities for which a history of certain attributes is kept like Project, Experiment, Lot, Wafer) look very similar and therefore require no individual descriptions. Only the experiment form will be outlined, the other entity maintenance modules will be skipped. The other tabs of the data maintenance modules will be described in the following descriptions.

All forms contain a dirty form checker and the changes are only saved to the persistence layer, if the ”Save” button is pressed to actually save all modified data. When navigating away from a form with changed data, the user will be notified and requested for a confirmation to leave with or without saving.

An example for an entity form of a tracked entity is shown in figure 6.14. It presents the Relation Explorer (described below) on the left and the ”General”



Experiment: IMS_PD.00028 - MEMS development external to CK

Entity Information

Number / Name: IMS_PD.00028 | MEMS development external to CK

Info: Created on 01.11.2004 03:26:30 by Dirk Ortloff
Edited on 03.02.2006 04:25:44 by PROMENADE Administrator

Description: MOnitor MEMS activities relevant to the company but external to the direct activities.

Start User / Date:

End User / Date:

Purpose: MOnitor MEMS activities relevant to the company but external to the direct activities.

Conclusion:

Suggested Wafer No. 0

Entity Meta Data

Analysis:

Analysis	Comment

Keywords:

Category	Keyword

Entity Status Information

Priority: - | No Comment Entered (1 more history items attached to this entity)

Milestone:

Status: Active | No Comment Entered (1 more history items attached to this entity)

Follow Up Required? □

Follow Up Summary:

User: Dirk Ortloff Groups:

10.0.0.85:1099

Figure 6.14.: Experiment Overview

tab of the experiment data maintenance module on the right. When opening a tracked entity, a screen with all detailed information will appear. If the opened entity is not locked by another user or the same user (opened but not closed before the current user) and the user has sufficient rights, it is possible to modify all fields of the entity. If the entity is locked, the forms are opened in read-only mode. It is possible to change all entity information like start date, start user, end date, end user and description. It is also possible to add, remove or change support information like the attached Analysis and Keywords as shown on the main screen. The lists can be manipulated with the buttons next to the list. The main screen also shows the latest values set for the tracked attributes Priority, Milestone and Status. To view or edit the history of these items the manage icon next to the current item needs to be selected. At last it is possible to specify whether a follow up is required, and what the comments are.

6.5.2. Concept additions

This part describes conceptual extensions added to the overall concepts and their implementations into the EVERIST system. Besides these conceptual additions a couple of new features are implemented into the EVERIST system to ease the usage of it. These features are explained in the next section 6.5.3.

Version & Document management

As introduced in section 6.4.2 about the PDTS system architecture, the document referencing via an URL can be considered a weakness of the technique chosen for the first concept implementation. This is due to the fact that referencial integrity can not be guaranteed because the URLs are not stable against deletion or renaming. Because of that the EVERIST implementation changed the approach to ensure that documents used in the system are always accessible.

For that a couple of possible mechanisms exist. One approach is to copy the files as a Binary Large OBject (BLOB) into the relational database. While this approach ensures the complete and consistent system back-up and referential integrity, it has the drawback of increasing the database size tremendously. Another mechanism is to always copy the files into a standard document management system (DMS) like Documentum [Doc05]. From there the documents could be referenced via the globally unique identifier the DMS system uses to manage the files. This approach fulfils all requirements but has the drawbacks of potentially inconsistent backup and that another, potentially proprietary and expensive system is used. A third way to ensure the referential integrity is similar to the DMS approach but uses a version control system. This approach has the drawback of potentially inconsistent backups as well but can use a free version control system like CVS [Fre05b] or Subversion [Sub05]. Additionally this approach has the advantage to manage several versions of the same file with a potentially lower footprint. This is because version control systems use mechanisms to store only differences between the versions of the file. This is very efficient for text files, for binary files the advantage is lower. But newer systems, like Subversion, handle the creation of differences of binary files with improved efficiency.

For the EVERIST system it has been decided to use the approach of accessing a version control sever. This is because of lower costs and the possibility to manage several versions of the same file efficiently. As a basis for the implementation the Subversion repository system was selected. The reasons were that it can operate server based, is more advanced than CVS, especially the difference methods for binary files are more efficient, and that it offers a portable JAVA interface.

The handling of the documents is separated out from the rest of the EVERIST system implementation into an own document management component. This has the advantage that the exchange of the Subversion system with a different system is much easier to achieve. The document management component provides the same services for all the other PROMENADE components and is not restricted to the EVERIST system. The document management component is used by the Design Environment and the Simulation Environment to store files associated directly with the process design or resulting simulation output. The documents are stored on a configurable server providing a local or remote Subversion repository. The interface between the document management component and the Subversion repository server is handled by a 100% native open source java library called JavaSVN. This removes the need for a platform specific implementations for interfacing with the Subversion repository.

The Document management uses two data persistence layers to store document information.

Database Meta-information about the file is stored in the relational database for faster indexing, searching etc. This information consists of descriptive information about the file itself and includes e.g. a description, unique identifiers, creation date and modification date.

Subversion The actual files are stored in the Subversion repository managing all the files and keeping a version history inside the files.

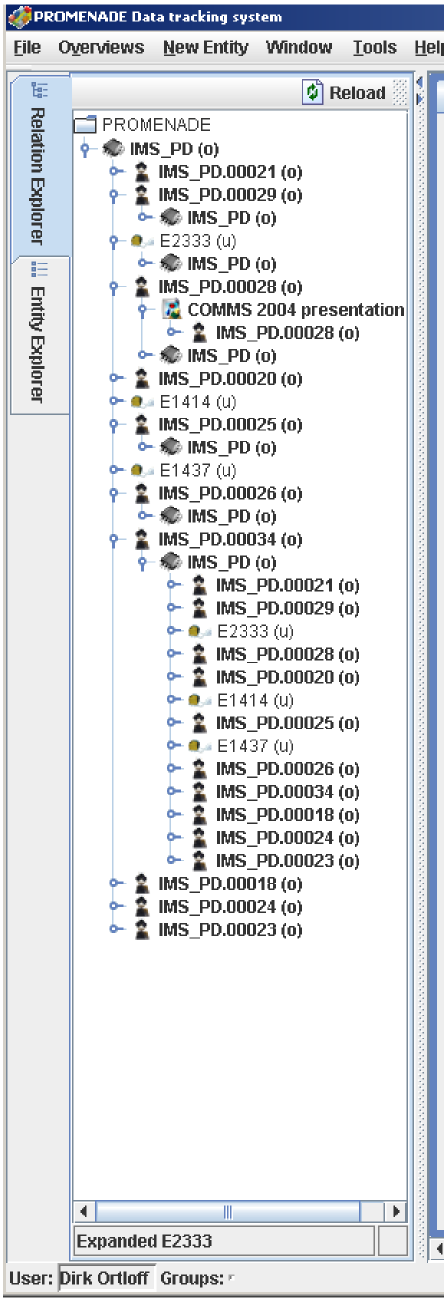
Navigation & Retrieval

Because the systematic collection of data is only one side of the medal the Navigation and Retrieval of/from the information storage and its different nodes is a major necessity. The system supports the information extraction and browsing in several different ways. This especially allows the assessment of the data from diverse perspectives as described in section 6.3.1. To allow this, several possibilities to navigate and retrieve data have been established and will be described from their conceptual perspectives in the following sections.

Relation Explorer The relations between the different entities are a major part of the captured knowledge. They can be used to explore the dataset as well. To allow such a navigation method the Relation Explorer has been integrated into the system. Following the relations from an arbitrary node allows to navigate through the information network. Because the relations between the entities can be directed or undirected (related in both directions; see section 7.1.1) it is possible to traverse the network in loops. The Relation Explorer starts from a list of projects, to have the highest entry point in the development hierarchy, and allows to explore their relations recursively. Expanding the folder of a project delivers all entities directly related to that project ordered by the different entity types. All entities retrieved can then be expanded to explore the network further in a file explorer fashion. Theoretically there is no limit for the deepness of the relationships because of the potential circular references. Every listed entity can be opened by the ”Open” option of the context menu. An example of the Relation Explorer is presented in figure 6.15. Via the context menu of the Relation Explorer new entities can be created as well. By selecting the ”Create new related Entity” option a new related entity of the selected type will be created. This opens the new entity creation wizard of the selected type described below.

Entity Explorer Another way to navigate through the information in the system is using the Entity Explorer shown in figure 6.16. The Entity Explorer behaves similar to the Relation Explorer but all entities are grouped by type as entry point. The behaviour for the listed entities are exactly the same as in the Relationship Explorer implying that recursive and unlimited deepness might occur. Opening the maintenance forms of the entities and the creation of new related entities is possible in the same way as in the Relation Explorer.

Relation graphing Another possibility to explore the relations of an arbitrary node is the graphical representation of the item with its relations. Here it should be possible to set the depth of the relation recursion, distance from the node in question, and to limit the presented entity types to a certain set. This approach has been implemented into the EVERIST system and an example of one possible graphical representation is given in figure 6.17. It presents the relations of the project ”IMS_PD” with a depth of two but limiting the presentation to experiments, lots, wafers and artefacts. Other entity types are not shown. The configuration for the drawing can be specified in the top right corner, with depth and types, and in the bottom right corner the corresponding relation explorer is shown. The colouring of the nodes in the graphic pane rep-

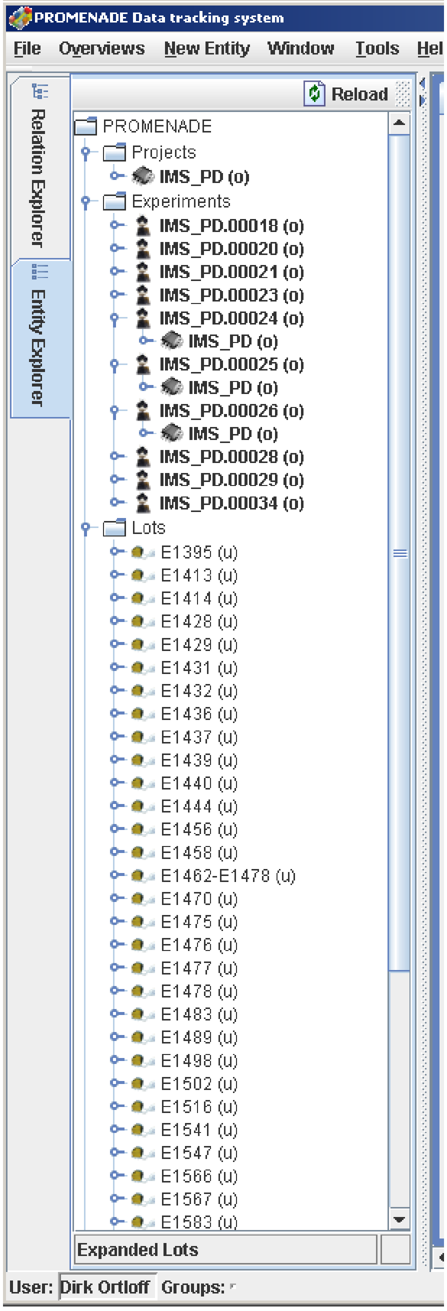

PROMENADE Data tracking system
File Overviews New Entity Window Tools Hel
Relation Explorer
Entity Explorer
Reload

```
PROMENADE
  IMS_PD (o)
    IMS_PD.00021 (o)
    IMS_PD.00029 (o)
      IMS_PD (o)
    E2333 (u)
      IMS_PD (o)
    IMS_PD.00028 (o)
      COMMS 2004 presentation
        IMS_PD.00028 (o)
      IMS_PD (o)
    IMS_PD.00020 (o)
    E1414 (u)
    IMS_PD.00025 (o)
      IMS_PD (o)
    E1437 (u)
    IMS_PD.00026 (o)
      IMS_PD (o)
    IMS_PD.00034 (o)
      IMS_PD (o)
        IMS_PD.00021 (o)
        IMS_PD.00029 (o)
        E2333 (u)
        IMS_PD.00028 (o)
        IMS_PD.00020 (o)
        E1414 (u)
        IMS_PD.00025 (o)
        E1437 (u)
        IMS_PD.00026 (o)
        IMS_PD.00034 (o)
        IMS_PD.00018 (o)
        IMS_PD.00024 (o)
        IMS_PD.00023 (o)
    IMS_PD.00018 (o)
    IMS_PD.00024 (o)
    IMS_PD.00023 (o)
```

Expanded E2333

User: Dirk Ortloff Groups:

Figure 6.15.: Relation Explorer

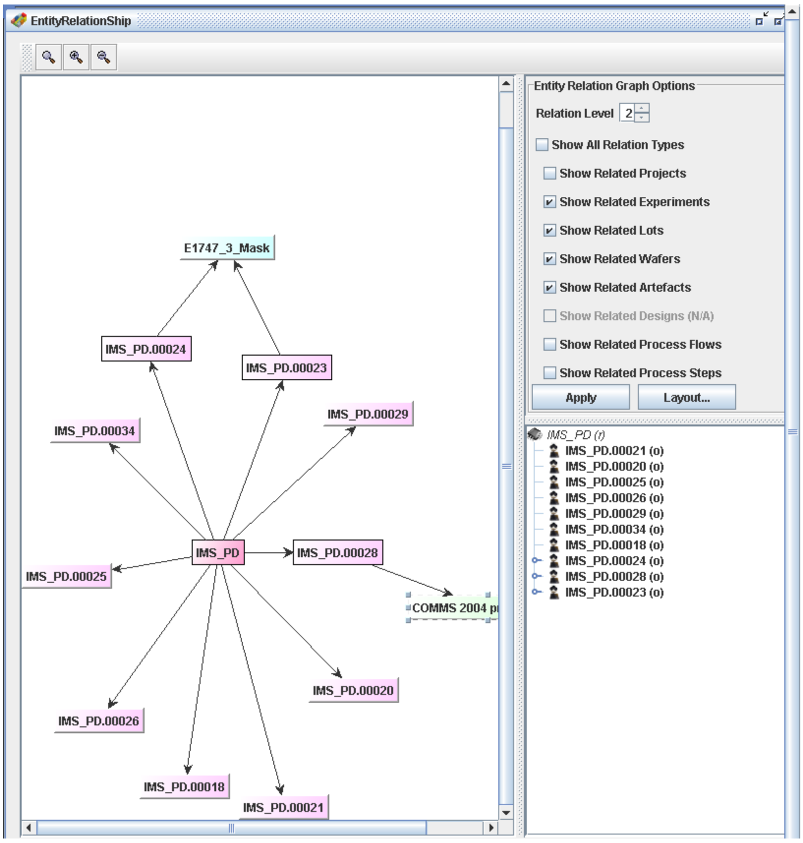

PROMENADE Data tracking system
File Overviews New Entity Window Tools Hel
Relation Explorer
Entity Explorer
Reload

```
PROMENADE
  Projects
    IMS_PD (o)
  Experiments
    IMS_PD.00018 (o)
    IMS_PD.00020 (o)
    IMS_PD.00021 (o)
    IMS_PD.00023 (o)
    IMS_PD.00024 (o)
      IMS_PD (o)
    IMS_PD.00025 (o)
      IMS_PD (o)
    IMS_PD.00026 (o)
      IMS_PD (o)
    IMS_PD.00028 (o)
    IMS_PD.00029 (o)
    IMS_PD.00034 (o)
  Lots
    E1395 (u)
    E1413 (u)
    E1414 (u)
    E1428 (u)
    E1429 (u)
    E1431 (u)
    E1432 (u)
    E1436 (u)
    E1437 (u)
    E1439 (u)
    E1440 (u)
    E1444 (u)
    E1456 (u)
    E1458 (u)
    E1462-E1478 (u)
    E1470 (u)
    E1475 (u)
    E1476 (u)
    E1477 (u)
    E1478 (u)
    E1483 (u)
    E1489 (u)
    E1498 (u)
    E1502 (u)
    E1516 (u)
    E1541 (u)
    E1547 (u)
    E1566 (u)
    E1567 (u)
    E1583 (u)
```

Expanded Lots

User: Dirk Ortloff Groups:

Figure 6.16.: Entity Explorer



Entity Relation Graph Options

Relation Level 2

☐ Show All Relation Types

☐ Show Related Projects

☑ Show Related Experiments

☑ Show Related Lots

☑ Show Related Wafers

☑ Show Related Artefacts

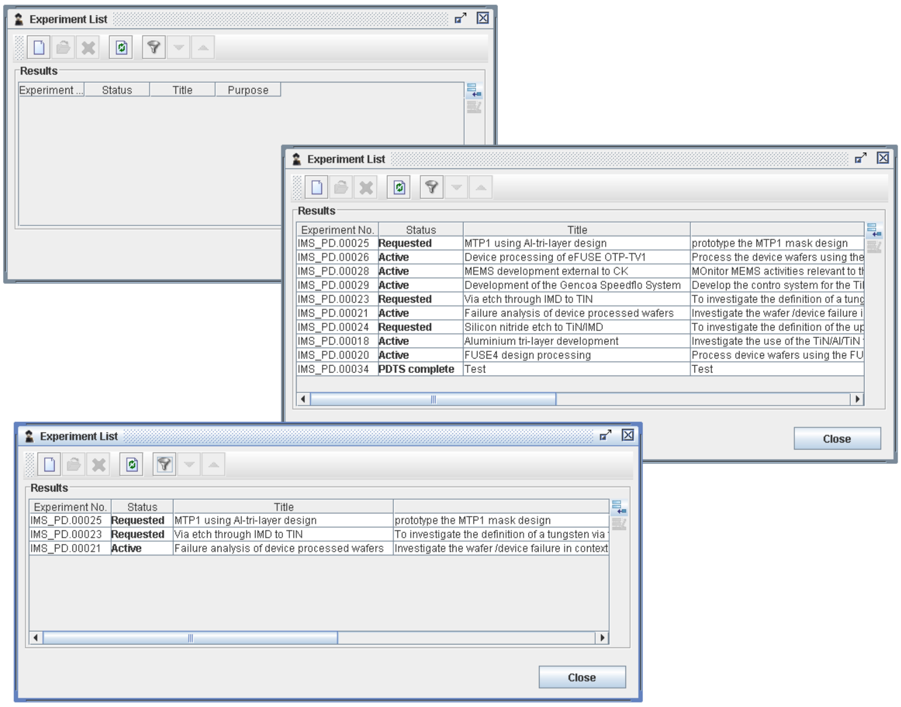
☐ Show Related Designs (N/A)

☐ Show Related Process Flows

☐ Show Related Process Steps

IMS_PD (r)
- IMS_PD.00021 (o)
- IMS_PD.00020 (o)
- IMS_PD.00025 (o)
- IMS_PD.00026 (o)
- IMS_PD.00029 (o)
- IMS_PD.00034 (o)
- IMS_PD.00018 (o)
- IMS_PD.00024 (o)
- IMS_PD.00028 (o)
- IMS_PD.00023 (o)

Figure 6.17.: Relation graph of a project



Experiment ...	Status	Title	Purpose

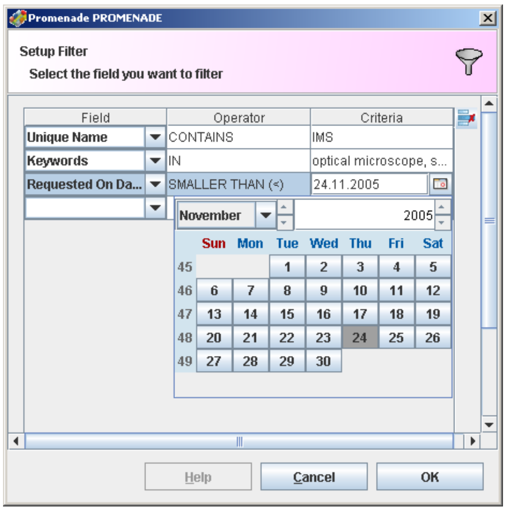
Experiment No.	Status	Title	
IMS_PD.00025	**Requested**	MTP1 using Al-tri-layer design	prototype the MTP1 mask design
IMS_PD.00026	**Active**	Device processing of eFUSE OTP-TV1	Process the device wafers using the
IMS_PD.00028	**Active**	MEMS development external to CK	MOnitor MEMS activities relevant to th
IMS_PD.00029	**Active**	Development of the Gencoa Speedflo System	Develop the contro system for the Ti
IMS_PD.00023	**Requested**	Via etch through IMD to TIN	To investigate the definition of a tung
IMS_PD.00021	**Active**	Failure analysis of device processed wafers	Investigate the wafer /device failure i
IMS_PD.00024	**Requested**	Silicon nitride etch to TiN/IMD	To investigate the definition of the up
IMS_PD.00018	**Active**	Aluminium tri-layer development	Investigate the use of the TiN/Al/TiN
IMS_PD.00020	**Active**	FUSE4 design processing	Process device wafers using the FU
IMS_PD.00034	**PDTS complete**	Test	Test

Experiment No.	Status	Title	
IMS_PD.00025	**Requested**	MTP1 using Al-tri-layer design	prototype the MTP1 mask design
IMS_PD.00023	**Requested**	Via etch through IMD to TIN	To investigate the definition of a tungsten via
IMS_PD.00021	**Active**	Failure analysis of device processed wafers	Investigate the wafer /device failure in context

Figure 6.18.: Experiment Overview

resent the different entity types and by clicking on the nodes the entities can be opened in data maintenance mode inside a new window.

Entity Overviews To open a certain entity without exploring the relations or selecting by entity type, the entity overviews can be used. Each entity type (Project, Experiment, Lot, etc.) has its own overview to list all entities of that type. Additionally the list can be filtered with diverse criteria, described below. In comparison to the Entity Explorer described above, the overviews present a certain set of preview data to more easily select the right entity. The overviews are sortable by the different attributes presented in the overview. They are accessible from the ”Overviews” menu and initially open the overview without showing any entity. This is presented in the top background of figure 6.18 for the experiment overview. The list is empty to allow setting filters (see below) before retrieving the list. The reason is that the full list of one entity type is potentially long and might take time to load. The window to the right of the figure presents the list of all experiment entities currently in the system where the current user has at least read access to. The full list can be retrieved by pressing the reload button without assigning a filter. To get all details of a specific experiment the entity can be opened by double clicking on that entity or via the ”Open” option of the context menu. The entity data management is described in more detail in section 6.5.1.



Promenade PROMENADE

Setup Filter

Select the field you want to filter

Field	Operator	Criteria
Unique Name	CONTAINS	IMS
Keywords	IN	optical microscope, s...
Requested On Da...	SMALLER THAN (<)	24.11.2005

November 2005

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
45			1	2	3	4	5
46	6	7	8	9	10	11	12
47	13	14	15	16	17	18	19
48	20	21	22	23	24	25	26
49	27	28	29	30			

Figure 6.19.: Filter

Filters To reduce the set of results retrieved in the different overviews, it is possible to define a Filter. Each overview is fitted with a filter button. Each filter consists of three parts as shown in figure 6.19 which are (1) the Filter Field, (2) the Operator and (3) the Criteria. Each entity can be filtered on all fields of the entity. Per filter field the user can choose an Operator from a predefined set of applicable ones for this field type. The Operator is part of the logical expression and defines what should happen with the Criteria. When editing a filter, the correct criteria editor is shown for the selected field. When filtering on status, it is required to select from status selector dialogue to select one or more status. The selection of several status for one filter acts as a logical ″OR″ operator. Figure 6.19 shows the date field selector open.

By adding more filters, the list of retrieved entities will become shorter because the different rows are used conjunctively. Together with the operators and their negations it provides a powerful tool to create customized overviews. The filters can be considered as a kind of searching functionality as well. The example presented in figure 6.19 filters experiments on the combination of the fields ″Unique Name″ containing the string ″IMS″ in conjunction with the collection of ″Keywords″ having ″optical microscope″ in it and having the field ″Requested on date″ begin smaller than the 24. November 2005. Applying this filter will retrieve all experiments fulfilling the combination of all these three criteria at once. The results for these settings are presented in the bottom window of figure 6.18. If no experiment exists fulfilling this set, the reloaded overview will be empty.

Forum system

To capture the discussions between engineers and the resulting knowledge, a forum system has been integrated into the EVERIST system. This functionality is similar to discussion forums on the Internet and substitutes the attachable texts of the PDTS system. The idea behind its addition is, to substitute email discussions with a systematic approach. Because these discussions represent an additional part of the knowledge, they should be captured inside the EVERIST system. The drawback of this new way of tracking the discussions is that it requires a change in the work procedures of the process engineers. This implies that the adoption of this new mechanism is expected to take some time.

Each entity in EVERIST has its own forum. A forum provides a way for users to discuss an entity in a more detailed and structured way. The forums have different discussion categories (e.g. Purposes, Conclusions, etc.) where the categories can be set up by the systems administrator in the basic configuration of the system. In fact the standard fields Purpose, Conclusion and Follow-up summary of all entities are default forum topics of the corresponding category. Therefore these categories can not be removed or changed and the categories have at least one default topic. To make these fields a topic has the advantage of additional discussion possibilities for these fields.

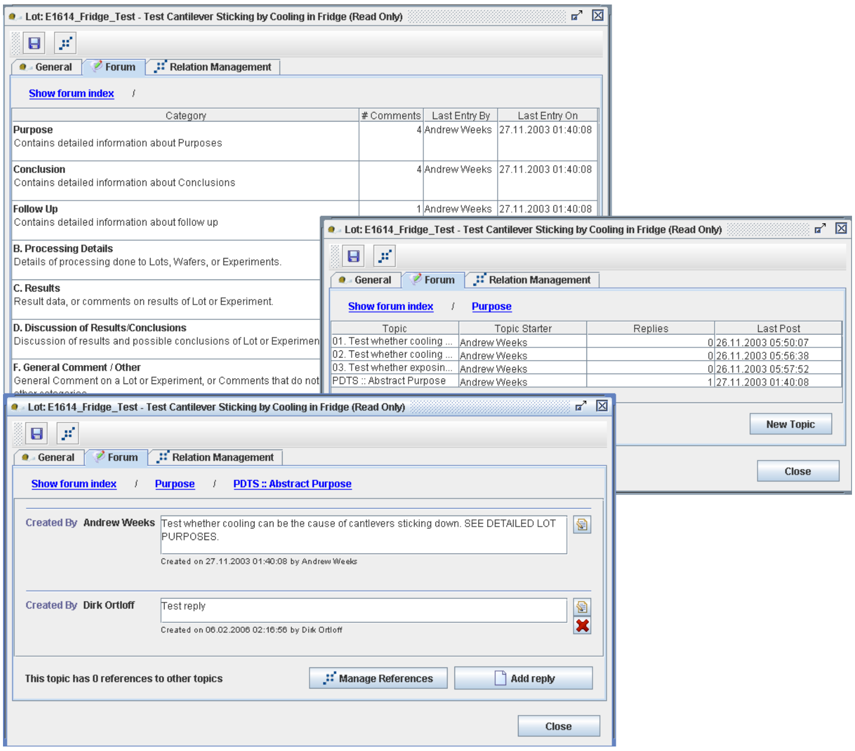
An overview of the different Forum categories is presented in the background of figure 6.20. Each forum category has topics. A topic is a discussion item that is created by a user and all other users, having access to this entity, can post their replies about that topic. A topic with a reply is shown in the foreground of 6.20. By these discussion possibilities the users can share their findings with the rest of the users. The overview of all topics of a forum category is presented in the middle picture of figure 6.20.

Additionally referencing between different topics is possible as described for the PDTS system in section 6.4.3. This allows to generate lists of related topics e.g. a list of purposes with corresponding conclusions.

Import&Export of data

As described in section 2.3.1 currently there is no electronic mechanism to transfer process IP. In the moment the transfer of the process knowledge is limited to documentations describing the process recipe and the resulting comparison data. This can be considered an issue because it causes difficult collection of data and tedious write-ups of recipes.

The EVERIST system addresses this issue in a flexible and automatised way. For that easy to use and generic mechanisms to import and export process data and information have been build into the system. A precondition for this export and import functionalities is a comprehensive intermediate format to



Lot: E1614_Fridge_Test - Test Cantilever Sticking by Cooling in Fridge (Read Only)

Category	# Comments	Last Entry By	Last Entry On
Purpose Contains detailed information about Purposes	4	Andrew Weeks	27.11.2003 01:40:08
Conclusion Contains detailed information about Conclusions	4	Andrew Weeks	27.11.2003 01:40:08
Follow Up Contains detailed information about follow up	1	Andrew Weeks	27.11.2003 01:40:08
B. Processing Details Details of processing done to Lots, Wafers, or Experiments.			
C. Results Result data, or comments on results of Lot or Experiment.			
D. Discussion of Results/Conclusions Discussion of results and possible conclusions of Lot or Experimen			
F. General Comment / Other General Comment on a Lot or Experiment, or Comments that do not			

Lot: E1614_Fridge_Test - Test Cantilever Sticking by Cooling in Fridge (Read Only)

Topic	Topic Starter	Replies	Last Post
01. Test whether cooling ...	Andrew Weeks	0	26.11.2003 05:50:07
02. Test whether cooling ...	Andrew Weeks	0	26.11.2003 05:56:38
03. Test whether exposin...	Andrew Weeks	0	26.11.2003 05:57:52
PDTS :: Abstract Purpose	Andrew Weeks	1	27.11.2003 01:40:08

Lot: E1614_Fridge_Test - Test Cantilever Sticking by Cooling in Fridge (Read Only)

Created By **Andrew Weeks** Test whether cooling can be the cause of cantlevers sticking down. SEE DETAILED LOT PURPOSES.

Created on 27.11.2003 01:40:08 by Andrew Weeks

Created By **Dirk Ortloff** Test reply

Created on 06.02.2006 02:16:56 by Dirk Ortloff

This topic has 0 references to other topics

Figure 6.20.: Forum overview showing all forum categories, the topic list of a category and a topic with reply

transfer the information. This was created by designing a Process Development and Tracking Markup Language (PDTML) based on the XML[1] standard. The PDTML is divided into three major parts and the schema definition is publicly available via a web site. The first part consists of the look-up data or master data like the definitions for status, milestones, priorities, keywords, etc. This part is followed by the actual domain data entities like experiment data, lot data, wafer data, etc. The last part contains relationship elements between the domain entities. For the artefacts and their belonging documents two possible export schemes are available. The first one stores the encoded documents inside the XML file making the XML files rather big and less open. Another mechanism is to store the documents as separate files (with or without history) and import them again into their related artefacts.

To export a domain model to XML and import complying XML data back into system an additional module has been added to the EVERIST system. It is written in Java and fitted into the rest of the system design. The import and export module has been added to the system using the the flexible system interface described in section 6.5.4. This implies that the import and export functionality can be added and removed from the EVERIST system with a plug-in approach.

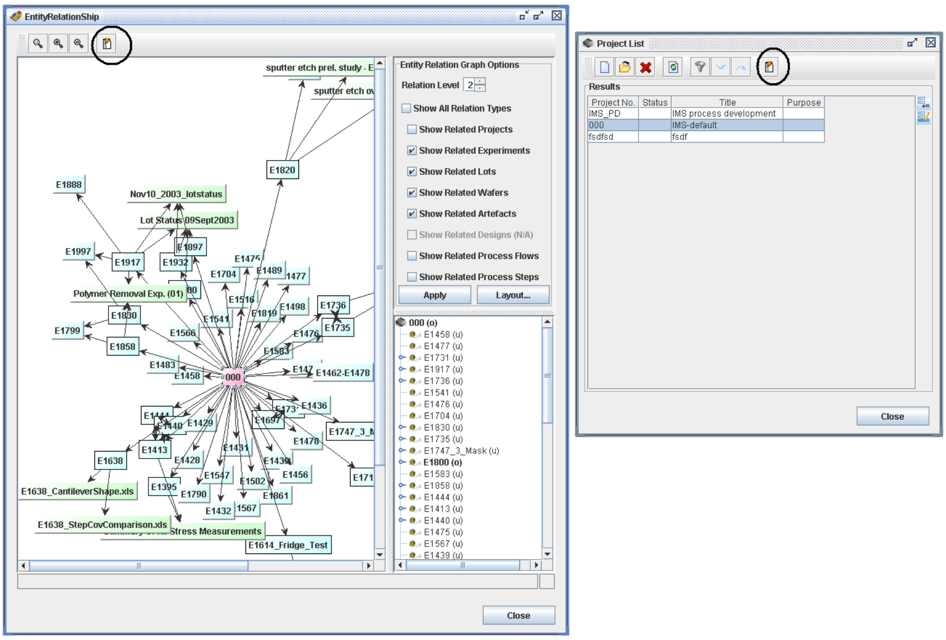
The exporting from the EVERIST system uses the visitor pattern to access and export data. The importing of the XML representation uses a combination of SAX[2] and JDOM[3]. SAX is used for parsing and validating whether an XML file complies with the PDTML schema language definition. Once the instance document is found to be valid it can be imported into the EVERIST system. This is done by using the JDOM API to create a JDOM object tree created by parsing the instance document. After the JDOM tree is built, it can be used to extract data and create domain objects.

The GUI integration of the import and export functionality has been done in a way to ease the selection for a selective export. Two ways of selecting are presented in figure 6.21. The left part of the figure shows the relation graph window and an additional button next to the zooming buttons (circled

[1] The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. It is a simplified subset of SGML. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. It is described in detail on its homepage [Wor04]

[2] SAX (http://www.saxproject.org) is currently maintained by the XML-DEV group (http://xml.org/xml/xmldev.shtml) and provides a way to validate an XML instance document to a schema, and parse this document. SAX was used as a way to validate the schema to the ”schema for schema” definition, and the instance document to the schema.

[3] JDOM is a Java-based document object model for XML that integrates with Document Object Model (DOM) and Simple API for XML (SAX) and uses parsers to build the document. It was designed specifically for the Java platform so that it can take advantage of its language features.



Entity Relation Graph Options

Relation Level 2

☐ Show All Relation Types
☐ Show Related Projects
☑ Show Related Experiments
☑ Show Related Lots
☑ Show Related Wafers
☑ Show Related Artefacts
☐ Show Related Designs (N/A)
☐ Show Related Process Flows
☐ Show Related Process Steps

Apply | Layout...

000 (o)
E1458 (u)
E1477 (u)
E1731 (u)
E1917 (u)
E1736 (u)
E1541 (u)
E1476 (u)
E1704 (u)
E1830 (u)
E1735 (u)
E1747_3_Mask (u)
E1800 (o)
E1583 (u)
E1858 (u)
E1444 (u)
E1413 (u)
E1440 (u)
E1475 (u)
E1567 (u)
E1439 (u)

Results

Project No.	Status	Title	Purpose
IMS_PD		IMS process development	
000		IMS-default	
fsdfsd		fsdf	

Figure 6.21.: Selective export from the relation graph

in). By pressing the export button only every entity and relation presented in the graph will be exported to an XML file. As explained in section 6.5.2 this information presentation allows to configure the elements shown in the graph. By adapting these presentation means, the export of the data can be limited as well. This allows to choose only data belonging to the intended subset e.g. export all data belonging to one project. The second way shown in figure 6.21 is integrated into the overviews. Here again the export button is integrated (circled in) and the selected entity can be exported with all its data and history but without any data of related entities.

IPR protection

Another important concept addition is the detailed handling of IPR (Intellectual Property Rights) protection. This is important because in this area the process knowledge is the key capital for the foundries and process IP companies. Protection is required between different business groups and between different users. The whole PROMENADE project addresses this need by building in a very strict mechanism of access protection. The project uses a shared user management component enabling the detailed management of user rights and roles. The Design, Tracking and Simulation Environments use this shared component as a common user and role database. The security management of each entity type is able to manage the access rights on a very low level. It is possible to protect every data set with regards to user groups and roles. Furthermore different levels of access have been realized (e.g. read, write, update,

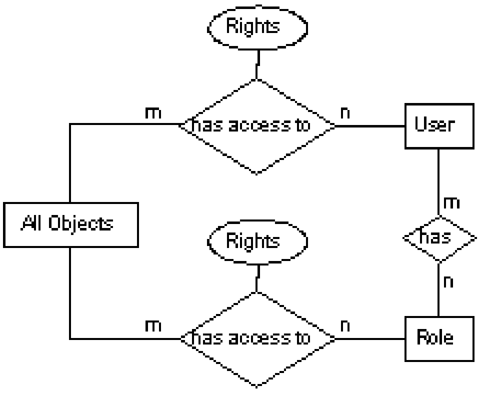


Figure 6.22.: User and role management in the User Management component

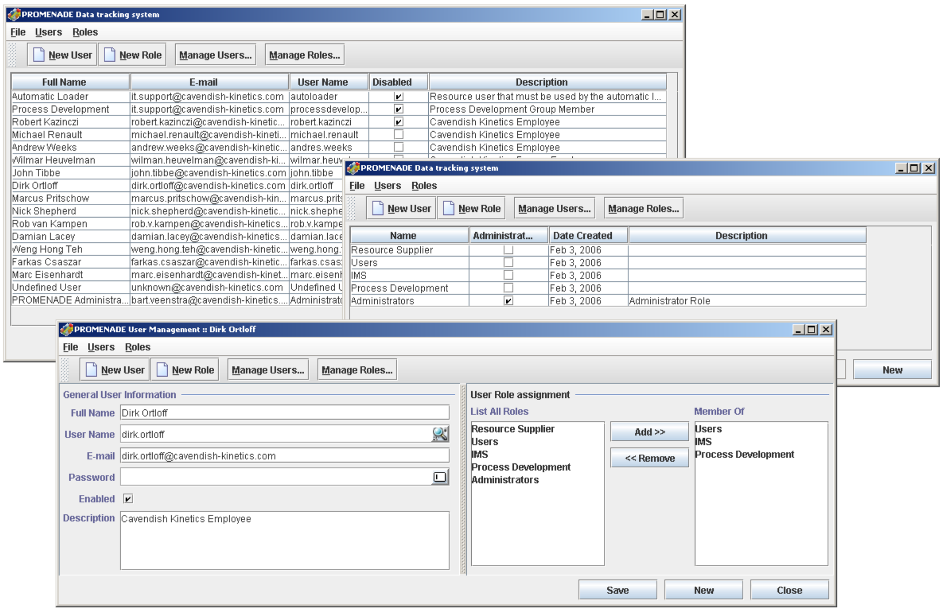
etc.; see below). With protection mechanisms of the relational database system itself this is difficult to achieve. But the database is used to handle access lists. These lists can be used by the application server to manage the access of the users logged in. Figure 6.22 shows an ER-diagram of the implementation of the user management.

Like the name indicates, the tables ”user” and ”role” contain the data regarding the different users of the system and their roles within the PROMENADE system. The management screens of the User Management component are shown in figure 6.23. The screenshot in the background shows a list of user and the screenshot in the middle shows the list of roles in the system. Every user can be assigned to different roles (e.g. Administrator, Process Development) as shown in the front screenshot for figure 6.23. For every object in the data management one access list for users and one list for roles is used to restrict the access. The rights of the users always overwrite the rights of the roles. This way it is possible to give the users of a certain group access to an object but restrict other users of the group from these access rights. Using this mechanism every object in the data management can have an individual set of rights. When a user or a role is deleted all rights of the user or role in the database are disabled. A role can only be deleted, when no users are assigned to the role.

To facilitate a fine grained rights management the User Management Component offers different levels of rights for the entities. The assignment of these rights is done via the Security Management tab of each entities form as shown for an experiment in figure 6.24. The security management distinguishes between the following right levels:

none: The user has no access to the entity. This implies that the entity is even invisible to the user.

read: The user can read the entity.



Full Name	E-mail	User Name	Disabled	Description
Automatic Loader	it.support@cavendish-kinetics.com	autoloader	☑	Resource user that must be used by the automatic l...
Process Development	it.support@cavendish-kinetics.com	processdevelop...	☑	Process Development Group Member
Robert Kazinczi	robert.kazinczi@cavendish-kinetic...	robert.kazinczi	☑	Cavendish Kinetics Employee
Michael Renault	michael.renault@cavendish-kineti...	michael.renault	□	Cavendish Kinetics Employee
Andrew Weeks	andrew.weeks@cavendish-kinetic...	andres.weeks	□	Cavendish Kinetics Employee
Wilmar Heuvelman	wilman.heuvelman@cavendish-ki...	wilmar.heuv		
John Tibbe	john.tibbe@cavendish-kinetics.com	john.tibbe		
Dirk Ortloff	dirk.ortloff@cavendish-kinetics.com	dirk.ortloff		
Marcus Pritschow	marcus.pritschow@cavendish-kin...	marcus.prits		
Nick Shepherd	nick.shepherd@cavendish-kinetic...	nick.shephe		
Rob van Kampen	rob.v.kampen@cavendish-kinetics...	rob.v.kampe		
Damian Lacey	damian.lacey@cavendish-kinetics...	damian.lace		
Weng Hong Teh	weng.hong.teh@cavendish-kinetic...	weng.hong.		
Farkas Csaszar	farkas.csaszar@cavendish-kinetic...	farkas.csas		
Marc Eisenhardt	marc.eisenhardt@cavendish-kinet...	marc.eisenh		
Undefined User	unknown@cavendish-kinetics.com	Undefined U		
PROMENADE Administra...	bart.veenstra@cavendish-kinetics....	Administrat		

Name	Administrat...	Date Created	Description
Resource Supplier	□	Feb 3, 2006	
Users	□	Feb 3, 2006	
IMS	□	Feb 3, 2006	
Process Development	□	Feb 3, 2006	
Administrators	☑	Feb 3, 2006	Administrator Role

General User Information

Full Name: Dirk Ortloff

User Name: dirk.ortloff

E-mail: dirk.ortloff@cavendish-kinetics.com

Password:

Enabled: ☑

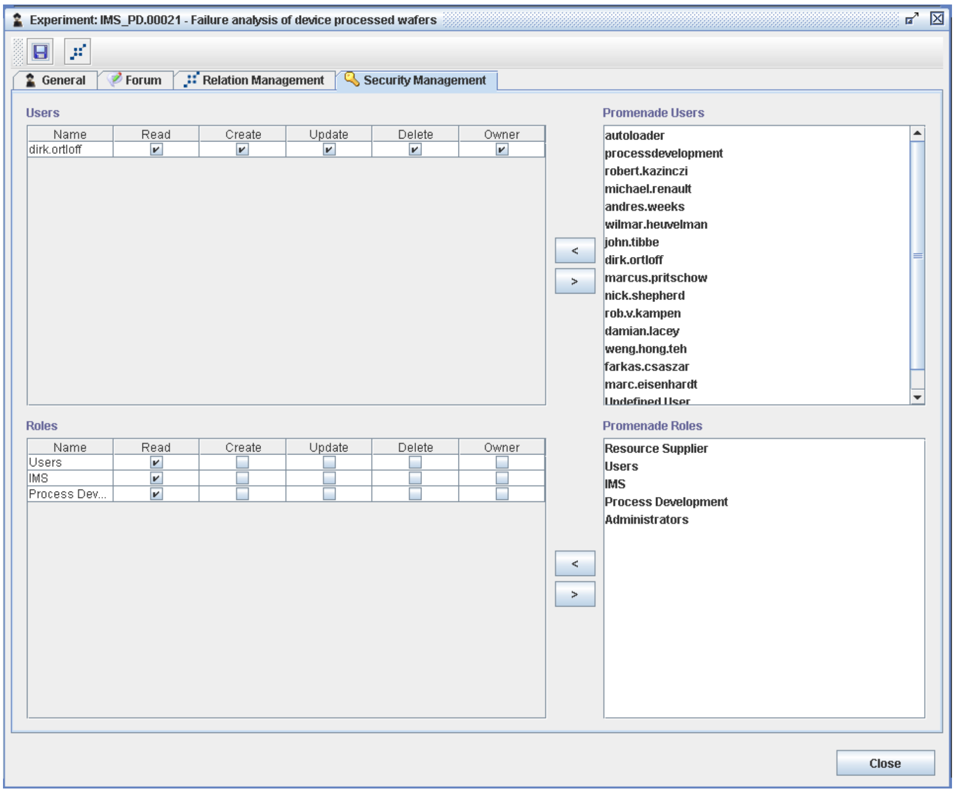
Description: Cavendish Kinetics Employee

User Role assignment

List All Roles: Resource Supplier, Users, IMS, Process Development, Administrators

Member Of: Users, IMS, Process Development

Figure 6.23.: Management of all users, of all roles and a users roles(back to front)



Experiment: IMS_PD.00021 - Failure analysis of device processed wafers
General
Forum
Relation Management
Security Management

Users

Name	Read	Create	Update	Delete	Owner
dirk.ortloff	☑	☑	☑	☑	☑

Promenade Users

autoloader
processdevelopment
robert.kazinczi
michael.renault
andres.weeks
wilmar.heuvelman
john.tibbe
dirk.ortloff
marcus.pritschow
nick.shepherd
rob.v.kampen
damian.lacey
weng.hong.teh
farkas.csaszar
marc.eisenhardt
Undefined User

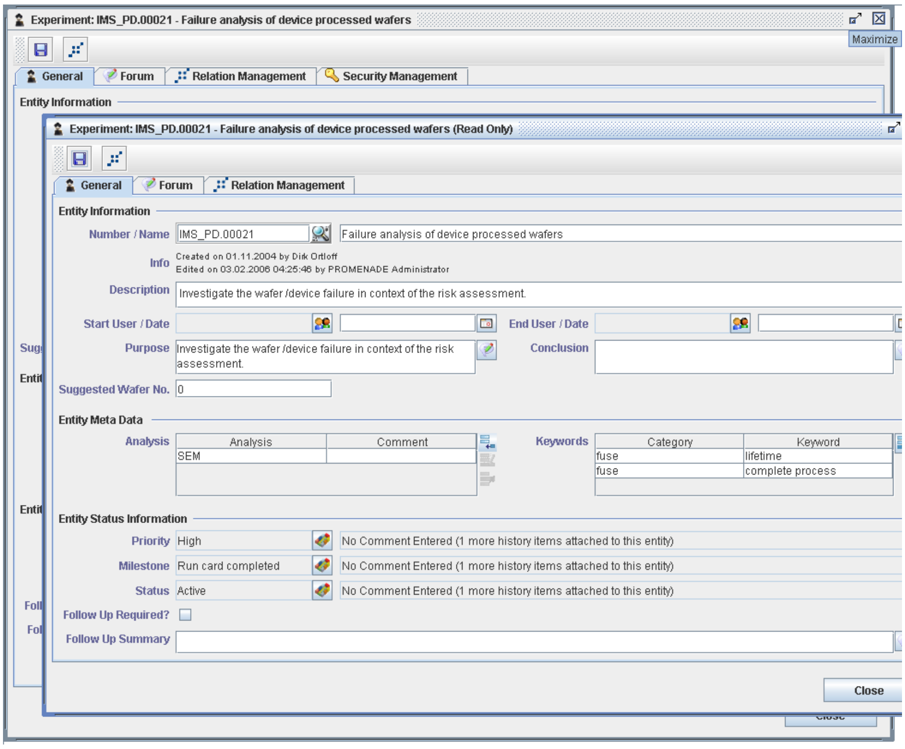
Roles

Name	Read	Create	Update	Delete	Owner
Users	☑	□	□	□	□
IMS	☑	□	□	□	□
Process Dev...	☑	□	□	□	□

Promenade Roles

Resource Supplier
Users
IMS
Process Development
Administrators

Figure 6.24.: Security Management tab of an Experiment



Experiment: IMS_PD.00021 - Failure analysis of device processed wafers (Read Only)

Entity Information

Number / Name: IMS_PD.00021 | Failure analysis of device processed wafers

Info: Created on 01.11.2004 by Dirk Ortloff
Edited on 03.02.2006 04:25:46 by PROMENADE Administrator

Description: Investigate the wafer /device failure in context of the risk assessment.

Start User / Date

End User / Date

Purpose: Investigate the wafer /device failure in context of the risk assessment.

Conclusion

Suggested Wafer No.: 0

Entity Meta Data

Analysis

Analysis	Comment
SEM	

Keywords

Category	Keyword
fuse	lifetime
fuse	complete process

Entity Status Information

Priority: High | No Comment Entered (1 more history items attached to this entity)

Milestone: Run card completed | No Comment Entered (1 more history items attached to this entity)

Status: Active | No Comment Entered (1 more history items attached to this entity)

Follow Up Required? □

Follow Up Summary

Figure 6.25.: Entity locking for the same user

create: The user can create new sub-entities (e.g.new subclasses) but cannot change the entity itself.

update: The user can change but not delete the entity.

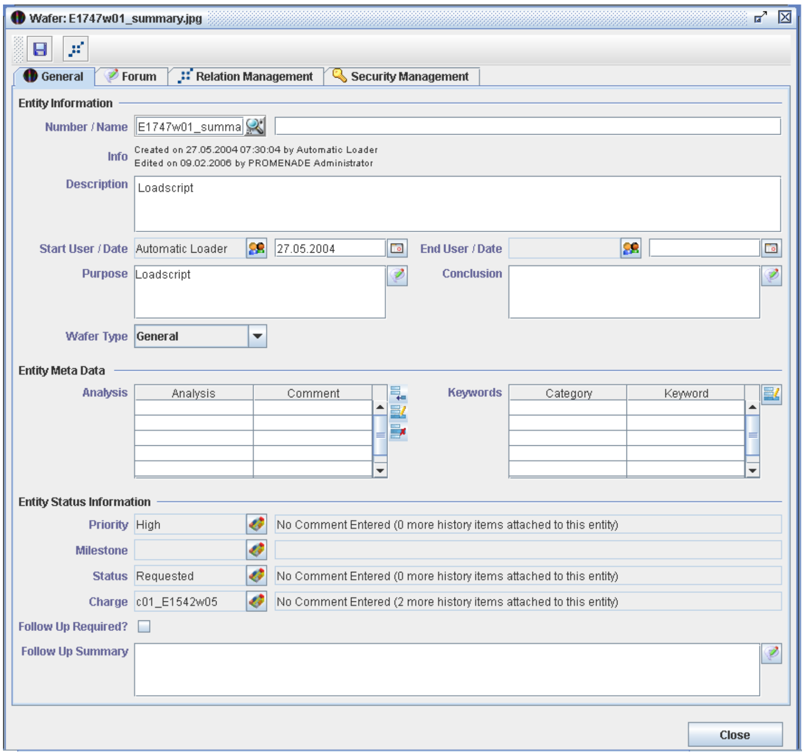
delete: The user can delete the entity.

owner: The user has all rights to the entity, including changing access rights.

Each level of rights includes the previous one, for example a user with update rights has also read and create rights.

Locking

Because the EVERIST system is a multi user system, it is necessary to avoid conflicts between simultaneously accessing users. Therefore a locking mechanism is used. If an entity is opened by a user with appropriate rights (at least update rights), the entity is locked for all other users or for simultaneous access of the same user. If a second access attempt is recognized, the system opens the second screen in read-only mode, independent of whether the same or a different user accesses the entity. An example of this mechanism is presented in figure 6.25 where the same user opens a second screen for the same entity. The second view, on top, is opened in read-only mode shown in the views title.



Entity Information

Number / Name: E1747w01_summa

Info: Created on 27.05.2004 07:30:04 by Automatic Loader
Edited on 09.02.2006 by PROMENADE Administrator

Description: Loadscript

Start User / Date: Automatic Loader 27.05.2004

End User / Date:

Purpose: Loadscript

Conclusion:

Wafer Type: General

Entity Meta Data

Analysis:

Analysis	Comment

Keywords:

Category	Keyword

Entity Status Information

Priority: High — No Comment Entered (0 more history items attached to this entity)

Milestone:

Status: Requested — No Comment Entered (0 more history items attached to this entity)

Charge: c01_E1542w05 — No Comment Entered (2 more history items attached to this entity)

Follow Up Required? □

Follow Up Summary:

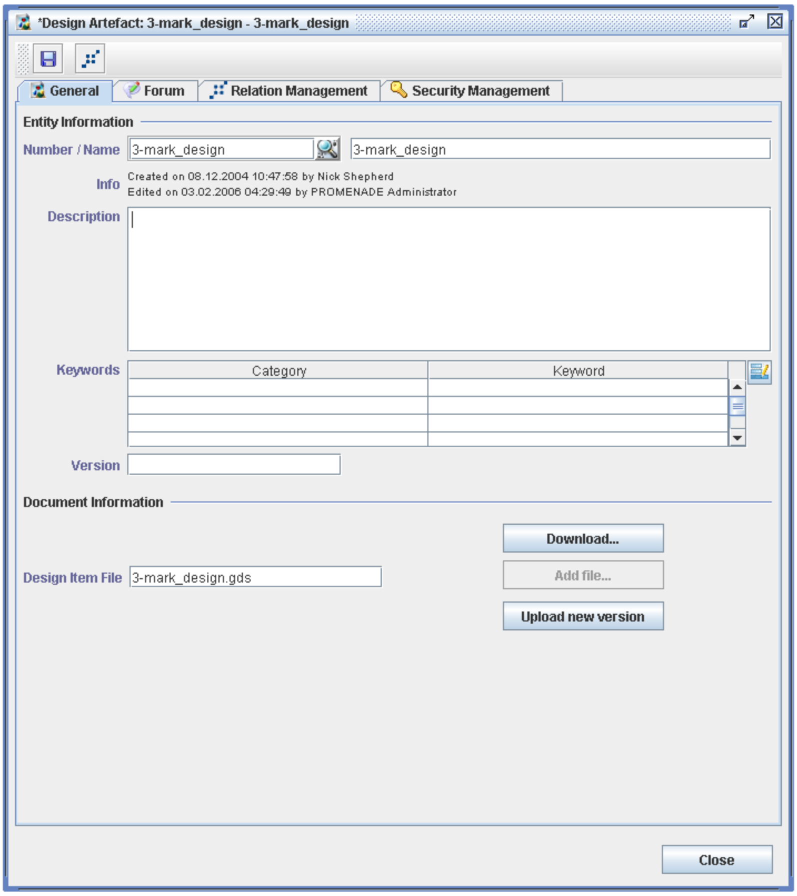
Figure 6.26.: Wafer form with the management of charges

All locks of a user are removed automatically about five minutes after the users connection to the server is lost. Additionally a user can remove his locks manually by selecting *Remove all Locks* from the *Tools* menu.

Management of Charges

As a matter of fact during the development and experimental verification activities a wafer does not necessarily constantly belongs to the same lot. It might be started within a lot, then the lot is split into two or more lots and these lots might be merged again afterwards. Another scenario is that a wafer temporarily is added to another lot and then back to its initial lot.

To flexibly allow the tracking of these reassignments of lot splits and merges the new concept of ”Charges” has been added to the EVERIST system. These splits and merges have been modelled by an additional, history tracked property called ”Charge” added to the wafer entities. This is shown in figure 6.26. It is very similar to e.g. the milestone history where the ”Charge No.” can be selected from a predefined set. This set can be edited in the basic configuration of the system and additionally while assigning and selecting a charge to a wafer. Commenting of the operators or engineers can be captured similarly to the other tracked properties. On top of that the time points of the changes of the ”Charge” belongings are tracked.



*Design Artefact: 3-mark_design - 3-mark_design

Entity Information

Number / Name: 3-mark_design | 3-mark_design

Info: Created on 08.12.2004 10:47:58 by Nick Shepherd
Edited on 03.02.2006 04:29:49 by PROMENADE Administrator

Description:

Keywords:

Version:

Document Information

Design Item File: 3-mark_design.gds

Figure 6.27.: Design entity

Additional entities

Design Entity As outlined in section 6.3.1 the mask design (lateral geometries defined in a set of masks) a wafer is manufactured with needs explicit modelling. The design entity has not been added to PDTS because the requirement to model the designs as a separate entity has been expressed rather late. Therefore it is only integrated into EVERIST. They are modelled in making them a special kind of artefact where only one document can be attached to. This document is version controlled as all other documents are but on top of the implicit version of the revision control system, an explicit version is kept in the meta-data. The explicit version is to identify a release of a mask set. Designs can be related to any other entity type, usually to wafers and lots. The relations can be used to express that a certain mask set is used to manufacture a wafer or lot. Figure 6.27 presents the ”General” tab of the design entity. It shows the different properties and the similarity to the normal artefacts. The Relation Management, Forum and Security tabs are similar to the other entities.

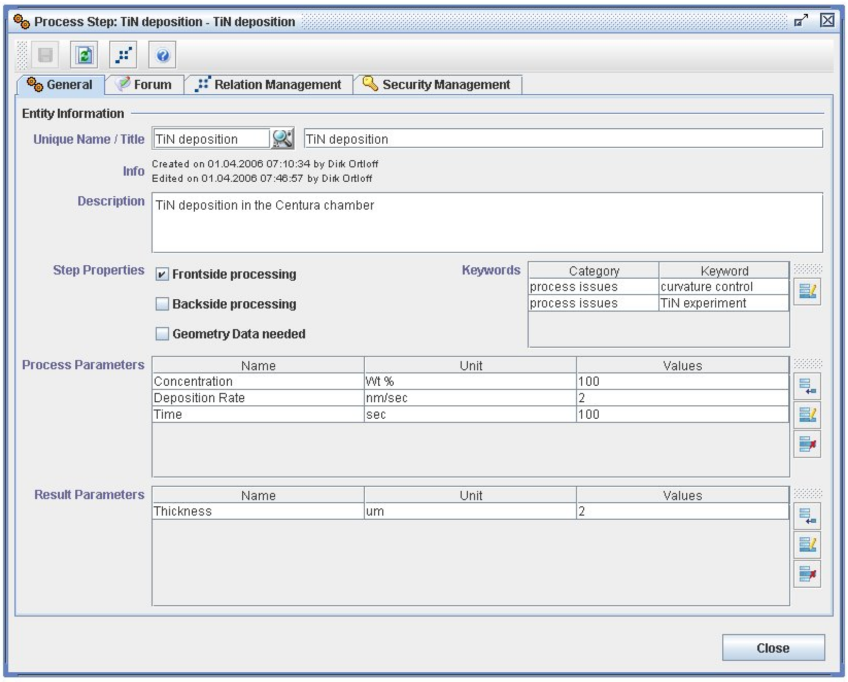
Process Step & Flow Entities The possibilities to manage the details of process steps and flows is essential to track the exact recipe applied during manufacturing a wafer. It is even important to be able to assess deviations between a planned recipe and a really applied recipe because the manufacturing might not comply hundred percent with the initially plan settings. Here the input from the fabrication equipment comes into play and the interfacing to MES systems could be very helpful. Only by the comparison between the details of the applied recipe and the verification result data cause-and-effect interdependencies can be derived.

The management of process steps and flows was contained in the initial concept and implementation of the PDTS system already. Due to the upcoming PROMENADE project these areas have not been developed further and the existing functionalities have been deactivated. The reasons were that the approach to design steps and flows proposed in the Design Environment are much more appealing and can be used in conjunction with the Tracking Environment.

In the PROMENADE system the Design Environment is integrated with the Tracking Environment and both components use the shared Parameter, Material and Unit manager component. The integration approach of Design and Tracking is that a process flow with its complete steps is transferred from the Design Environment to the Tracking Environment when a run card for the manufacturing is created. Another possibility is the manual transfer importing a selected process step or flow from the Design into Tracking. The designed recipe remains in the Design Environment and the recipe applied in the manufacturing line is tracked in the Tracking Environment. Here it is related together with the other experimental verification entities (experiment, lot, wafer, etc.). Therefore the deviating settings and each application of a recipe are captured in the Tracking Environment which keeps a back reference to the Design Environment as well. This back reference allows for comparisons between planned and real recipes.

Inside the Tracking Environment the recipe can be fully maintained, changed, commented, etc. For that the steps and the flows are modelled as separate entities and can be manipulated with the forms presented in figure 6.28 and 6.29 respectively. The first figure presents the Process Step maintenance form and the second the Process flow maintenance form. Both entity types contain the usual forum, relationship management and security tabs.

Assessment Reports The new entity type Assessment Report presented in figures 6.3, 6.5 and 6.6 has been added lately to the overall concept. The requirements are still under development and therefore will only be briefly described here. The main requirements for this new entity type arose from the needs to capture in detail all tests and assessments applied to a wafer or a part



Process Step: TiN deposition - TiN deposition

Entity Information

Unique Name / Title: TiN deposition | TiN deposition

Info: Created on 01.04.2006 07:10:34 by Dirk Ortloff
Edited on 01.04.2006 07:46:57 by Dirk Ortloff

Description: TiN deposition in the Centura chamber

Step Properties:
☑ Frontside processing
□ Backside processing
□ Geometry Data needed

Keywords

Category	Keyword
process issues	curvature control
process issues	TiN experiment

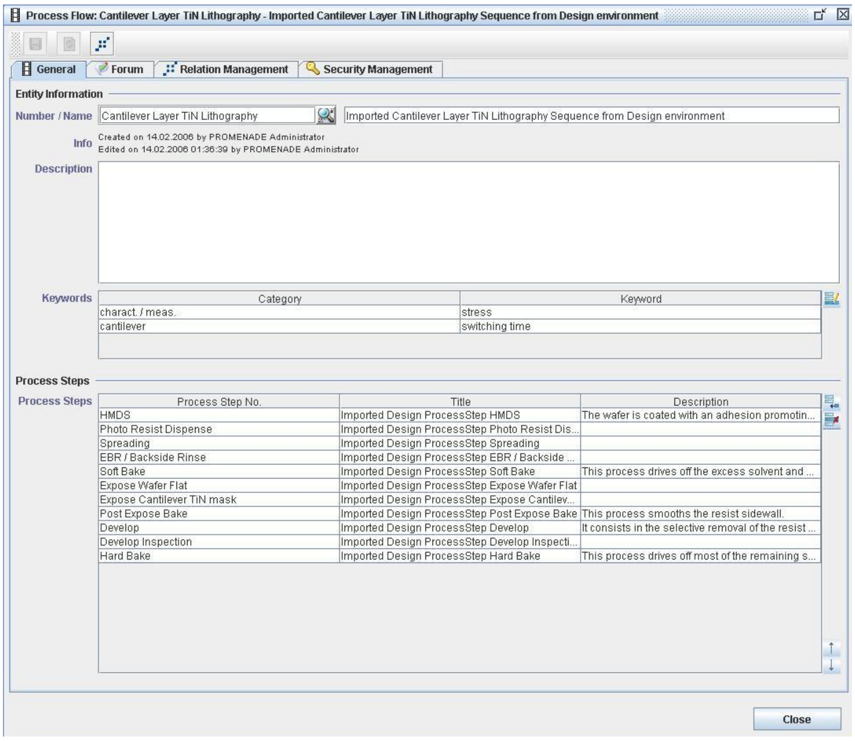
Process Parameters

Name	Unit	Values
Concentration	Wt %	100
Deposition Rate	nm/sec	2
Time	sec	100

Result Parameters

Name	Unit	Values
Thickness	um	2

Figure 6.28.: Process Step main screen



Process Flow: Cantilever Layer TiN Lithography - Imported Cantilever Layer TiN Lithography Sequence from Design environment

Entity Information

Number / Name: Cantilever Layer TiN Lithography | Imported Cantilever Layer TiN Lithography Sequence from Design environment

Info: Created on 14.02.2006 by PROMENADE Administrator
Edited on 14.02.2006 01:36:39 by PROMENADE Administrator

Description:

Keywords

Category	Keyword
charact. / meas.	stress
cantilever	switching time

Process Steps

Process Step No.	Title	Description
HMDS	Imported Design ProcessStep HMDS	The wafer is coated with an adhesion promotin...
Photo Resist Dispense	Imported Design ProcessStep Photo Resist Dis...	
Spreading	Imported Design ProcessStep Spreading	
EBR / Backside Rinse	Imported Design ProcessStep EBR / Backside ...	
Soft Bake	Imported Design ProcessStep Soft Bake	This process drives off the excess solvent and ...
Expose Wafer Flat	Imported Design ProcessStep Expose Wafer Flat	
Expose Cantilever TiN mask	Imported Design ProcessStep Expose Cantilev...	
Post Expose Bake	Imported Design ProcessStep Post Expose Bake	This process smooths the resist sidewall.
Develop	Imported Design ProcessStep Develop	It consists in the selective removal of the resist ...
Develop Inspection	Imported Design ProcessStep Develop Inspecti...	
Hard Bake	Imported Design ProcessStep Hard Bake	This process drives off most of the remaining s...

Figure 6.29.: Process Flow main screen

of a wafer. Especially failures and defects of devices and new, unexpected behaviours need to be captured. It should be possible to described these within summary test reports, failure reports and conclusion reports and link the different reports together. The reports should allow the detailed location of the observation points within lots, wafers, etc., allow to describe the detailed test conditions and div. further properties.

The realization of the entity forms will be based on the same approach of the existing entity forms. The ”General” tab will contain several new look-up types for e.g. report types, test equipments and test techniques and fields to describe the test conditions and results. Free text fields will be mapped into the forum to be able to discuss about the items and to be able to relate these items together. Within the ”Forum” a slightly extended forum mechanism will be realized to cater for possibilities to link the results to the corresponding lots, wafers etc. and collect the data in a matrix style for testing several locations within the same test sequence.

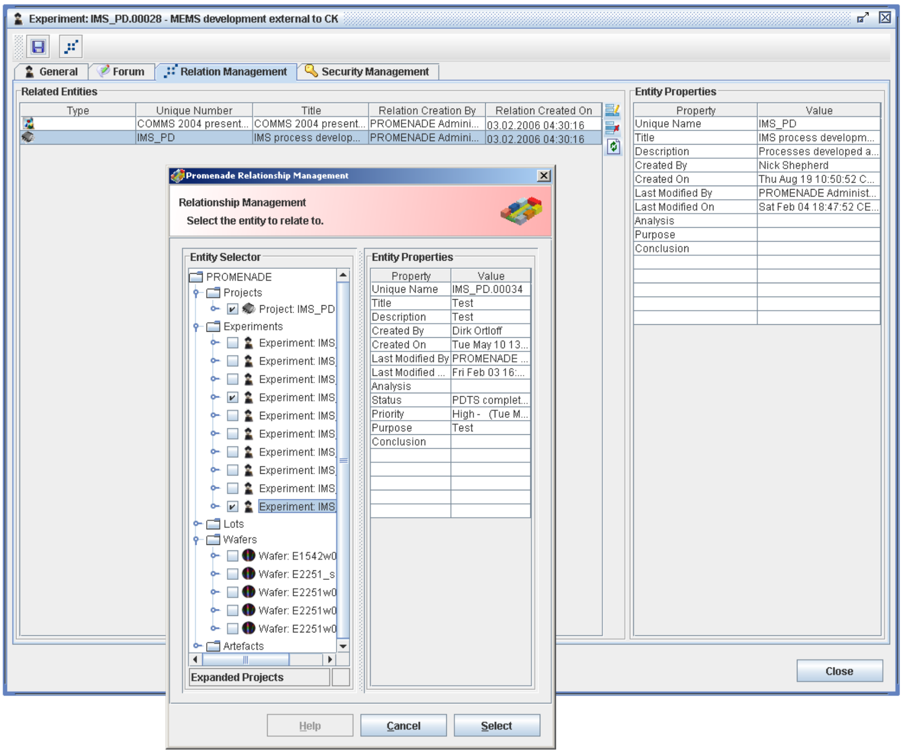
6.5.3. New and changed Features

The current section describes the added and changed features of the EVERIST system. Here only items are described which ease the usage of the system without significantly extending the underlaying concepts. The concept additions are described in the previous section 6.5.2. Detailed descriptions of the functionalities and their usage are documented in the PROMENADE manual [Ins05].

Relation Management

Relation management is a major part of the tracking system and the capturing of the experimental verification. Therefore it is important that the management is easy, efficient and intuitive. In PDTS the relation management was split into explicit management screens for each related entity pair(e.g. a screen to manage document to experiment relations, a screen to manage document to lot relations etc.). Therefore it was sometimes complicated and cumbersome to set all relations of one entity e.g. a document because several screens had to be used. This has been improved in the EVERIST system by using an improved user guidance.

The relationship management can be performed using the Relationship Management tab of the entity views. This tab is presented in the background of figure 6.30. It list all relations of the current entity together with some indicative data of the related entities. On the right side of the screen a preview with more details about the selected entity is presented. The management of the relations can be done by opening the relation management dialogue shown in the foreground of figure 6.30. This dialogue consists of an entity browser order-



Type	Unique Number	Title	Relation Creation By	Relation Created On
	COMMS 2004 present...	COMMS 2004 present...	PROMENADE Admini...	03.02.2006 04:30:16
	IMS_PD	IMS process develop...	PROMENADE Admini...	03.02.2006 04:30:16

Property	Value
Unique Name	IMS_PD
Title	IMS process developm...
Description	Processes developed a...
Created By	Nick Shepherd
Created On	Thu Aug 19 10:50:52 C...
Last Modified By	PROMENADE Administ...
Last Modified On	Sat Feb 04 18:47:52 CE...
Analysis	
Purpose	
Conclusion	

Relationship Management

Select the entity to relate to.

Entity Selector

- PROMENADE
  - Projects
    - ☑ Project: IMS_PD
  - Experiments
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☑ Experiment: IMS
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☐ Experiment: IMS
    - ☑ Experiment: IMS
  - Lots
  - Wafers
    - ☐ Wafer: E1542w0
    - ☐ Wafer: E2251_s
    - ☐ Wafer: E2251w0
    - ☐ Wafer: E2251w0
    - ☐ Wafer: E2251w0
  - Artefacts

Expanded Projects

Entity Properties

Property	Value
Unique Name	IMS_PD.00034
Title	Test
Description	Test
Created By	Dirk Ortloff
Created On	Tue May 10 13...
Last Modified By	PROMENADE ...
Last Modified ...	Fri Feb 03 16:...
Analysis	
Status	PDTS complet...
Priority	High - (Tue M...
Purpose	Test
Conclusion	

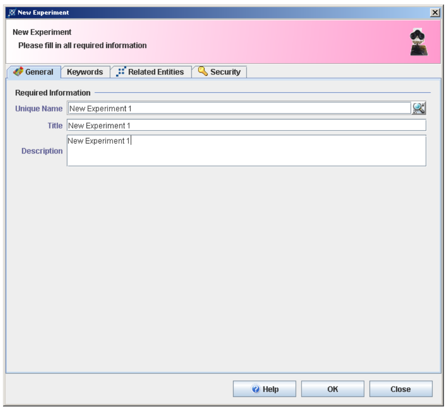
Figure 6.30.: Relation Management and the relation dialogue

ing the entities by the entity types and a preview window for the highlighted entity. The creation or removal of relationships between the edited entity and another entity can be done by ticking or unticking the boxes next to the entities. By navigating through the different entity types, all relations of an entity can be managed within one screen. Leaving the dialogue via the Select button changes all relationships set within the dialogue. This new way significantly improved the usability of the relationship management.

New entity wizards

To ease the creation of new entities, two types of wizards have been implemented guiding through the creation of new entities.

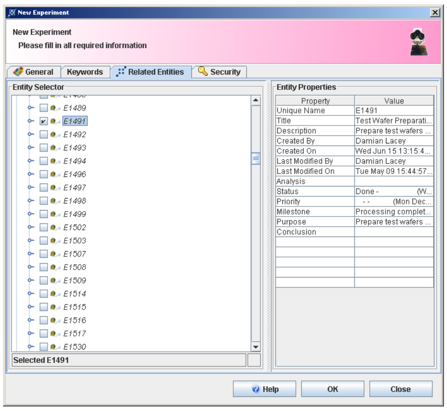
Single entity wizards The single entity wizards create one entity at a time. Each entity type has its own wizard creating a new entity with a minimum amount of data. The wizards can be started from the *New Entity* menu or via the context menu of the Relation and Entity Explorer. The wizards require to specify the unique name, a title and a description of the entity to create. The uniqueness of the name can be checked by the selecting the magnifying glass. These settings are done on the ”General” tab of the wizards as presented in figure 6.31. Additionally the user can specify the keywords he wants to attach



New Experiment
Please fill in all required information

Required Information
Unique Name: New Experiment 1
Title: New Experiment 1
Description: New Experiment 1

Figure 6.31.: General tab of the new experiment wizard



New Experiment
Please fill in all required information

Entity Selector
- □ E1489
- ☑ E1491
- □ E1492
- □ E1493
- □ E1494
- □ E1496
- □ E1497
- □ E1498
- □ E1499
- □ E1502
- □ E1503
- □ E1507
- □ E1508
- □ E1509
- □ E1514
- □ E1515
- □ E1516
- □ E1517
- □ E1530

Selected E1491

Entity Properties

Property	Value
Unique Name	E1491
Title	Test Wafer Preparati...
Description	Prepare test wafers ...
Created By	Damian Lacey
Created On	Wed Jun 15 13:15:4...
Last Modified By	Damian Lacey
Last Modified On	Tue May 09 15:44:57...
Analysis	
Status	Done - (W...
Priority	- - (Mon Dec...
Milestone	Processing complet...
Purpose	Prepare test wafers ...
Conclusion	

Figure 6.32.: Relations tab of the new experiment wizard

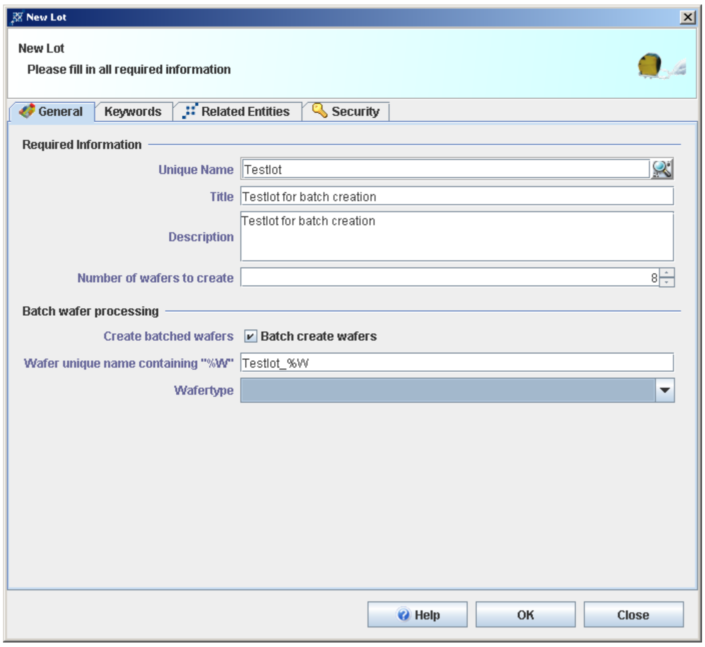
immediately to the new entity. This is performed on the second tab of the wizard. On the third tab, called ”Related Entities”, all relations can be selected. The tab contains the same Relationship Management browser like the regular Relationship Management and is presented in figure 6.32. The third tab allows to set the default security setting for the created entity.

Batch entity wizards The idea of the batch entity wizard is to allow to generate all wafers of a lot while creating a new lot with the lot creation wizard. This removes the subsequent manual creation of the wafers for a lot. If the option of batch creation is selected in the lot wizard, shown in figure 6.33, the system proposes a name for the wafers. This name can freely be changed. As an extension of the name, the system will add a suffix with a counter for the wafers with their numbers (shown in square brackets in the figure). Additional to the creation of the pure entities, all relations between the wafers and the lot are created. Besides this the batch lot creation wizard is similar to the other entity creation wizards.

Distinction of artefacts and documents

Another changed concept is the distinction between artefacts and documents. The PDTS system supports the concept of a document reflecting a single URL with meta-data. The URL in PDTS could be a direct link to a file or a link to a directory. This was useful to not include links to e.g. all SEM pictures, instead one document contained a link to one directory holding a set of pictures of one wafer.

A slightly modified approach has been implemented into EVERIST. Here a document or artefact-item always represents a single file and adds no meta-data to it. These files reside inside the Revision Control system as described



New Lot

Please fill in all required information

Required Information

Unique Name: Testlot

Title: Testlot for batch creation

Description: Testlot for batch creation

Number of wafers to create: 8

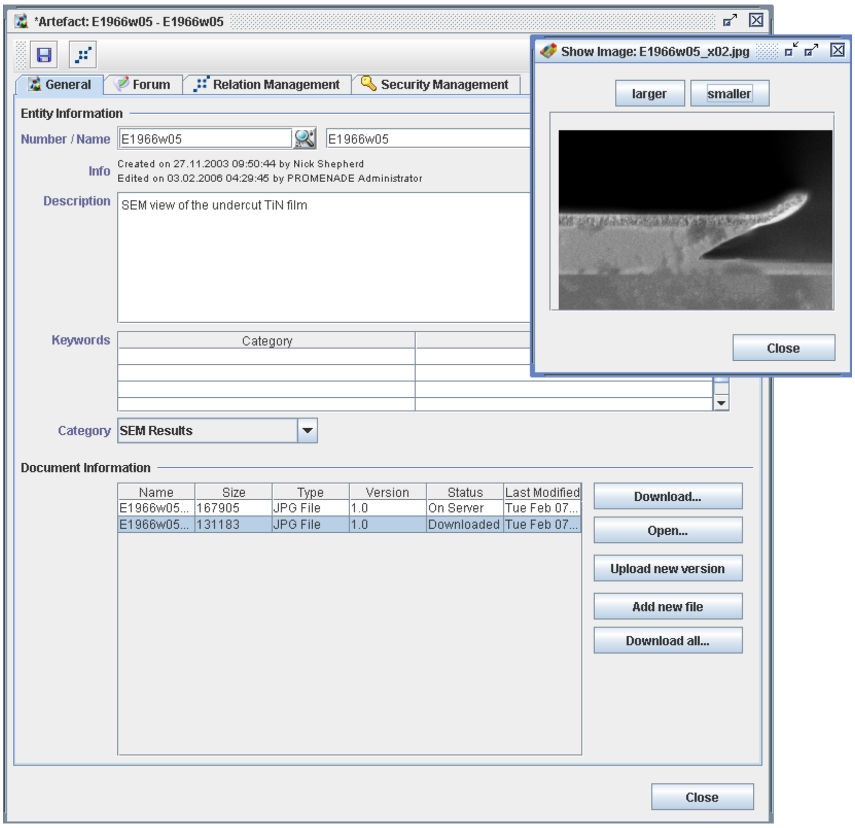
Batch wafer processing

Create batched wafers: ☑ Batch create wafers

Wafer unique name containing "%W": Testlot_%W

Wafertype:

Figure 6.33.: Lot wizard creating a full batch of ten wafers



Entity Information

Number / Name: E1966w05 E1966w05

Info: Created on 27.11.2003 09:50:44 by Nick Shepherd
Edited on 03.02.2006 04:29:45 by PROMENADE Administrator

Description: SEM view of the undercut TiN film

Keywords

Category	

Category: SEM Results

Document Information

Name	Size	Type	Version	Status	Last Modified
E1966w05...	167905	JPG File	1.0	On Server	Tue Feb 07...
E1966w05...	131183	JPG File	1.0	Downloaded	Tue Feb 07...

Figure 6.34.: Artefact management form

in section 6.5.2. The wrapper around documents are called artefacts and hold meta-data for a single or for multiple documents. With this concept it is possible to group arbitrary files, independent of the location, together into the abstract item called artefact as well as supporting the directory approach of PDTS. The document type of PDTS is as well moved to the artefacts, now being called artefact category. The artefact categories can be used in several places to group or select certain artefacts.

The artefact management form is presented in figure 6.34. It shows the meta-data of the artefact, its category and the list of documents or artefact-items belonging to the artefact. Via the different buttons on the right side it is possible to download the last version and previous versions of a document, to open the last version of a document into a preview or application, to upload a new version of a document, to add new documents to the artefact and to download all documents of that artefact. The open functionality is presented in the small inset picture showing the second document belonging to that artefact which is a SEM image of a structure.

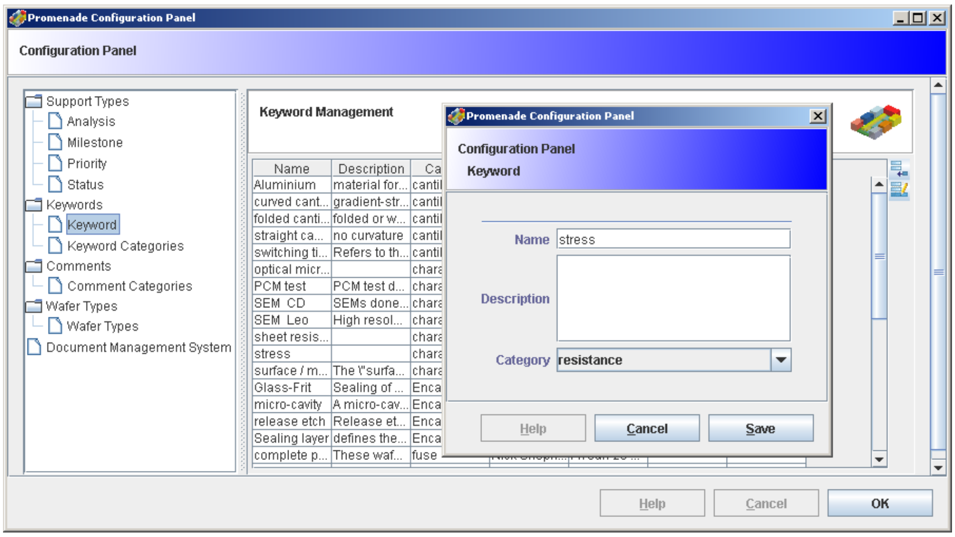
Loading into the system

The data loading mechanisms of PDTS have been removed from the core EVERIST system and have been integrated into the Back Annotation component. The reason for this decision was to make the whole loading approach more versatile and more generic to be able to load data into other PROMENADE components as well. Because the core task of the Back Annotation component is to shift data between different systems, the integration of the loading into that component gives a cleaner architecture. Anyhow, the Back annotation component is still closely related to the Tracking component. The new loading mechanisms are described in section 6.6.1.

Basic Configuration

For the EVERIST systems basic configuration an approach similar to the PDTS administration area has been used. Via the *Configuration* from the *Tools* menu the basic lookup types of the system can be configured. Here the different status, milestones, keywords, etc. can be configured. Figure 6.35 gives an overview about the different configuration possibilities.

The screen is separated into two different parts. On the left all configurable items are shown. On the right a list of all items as well as all the options of the current configuration type are shown. The figure presents the list of all keywords in the system. On the right side of the window a button bar can be found with all available options. In this case the options are ”New” and ”Edit”. The front window of the figure 6.35 shows the edit keyword dialogue.



Name	Description	Ca
Aluminium	material for...	cantil
curved cant...	gradient-str...	cantil
folded canti...	folded or w...	cantil
straight ca...	no curvature	cantil
switching ti...	Refers to th...	cantil
optical micr...		chara
PCM test	PCM test d...	chara
SEM CD	SEMs done...	chara
SEM Leo	High resol...	chara
sheet resis...		chara
stress		chara
surface / m...	The \"surfa...	chara
Glass-Frit	Sealing of ...	Enca
micro-cavity	A micro-cav...	Enca
release etch	Release et...	Enca
Sealing layer	defines the...	Enca
complete p...	These waf...	fuse

Name: stress

Description:

Category: resistance

Figure 6.35.: Configuration of Keywords

All other options in the configuration panel can be modified in a similar way. Therefore they will not be explained.

6.5.4. System architecture

This section discusses the EVERIST system architecture. Because the system is embedded into the PROMENADE environment, the architecture has been influenced by the PROMENADE architecture decision. For that the following subsection describes the general architecture possibilities and the section after that describes the conclusive architecture outline derived from these discussions.

Architecture possibilities

The architecture of a software system describes its structure and the communication infrastructure. In principle software systems can be divided roughly into two categories. Monolithic systems running only on a single machine and client-server systems, spread over several machines. For the PROMENADE environment the monolithic approach was ruled out because of the usage of existing foundation systems and the intended modular approach.

Several-Tier architectures Several-tier architectures can be classified as 2-tier, 3-tier or multi-tier models depending on their structure.

- 2-tier architectures
  The 2-tier architectures contain the application layer with the business logic, combined with the GUI, and the data layer. The advantage of this

kind of systems is that the communication to the data layer usually happens via a very well defined interface enabling the easy exchange of the data layer. The drawback is that the kernel of the program is still monolithic mixing the logic with the GUI.

- 3-tier architectures
  Compared to the 2-tier architecture the 3-tier architecture enables the splitting of the GUI (presentation layer) from the logic layer (application layer) where the GUI is executed on the client but the logic and the calculation intensive tasks are executed on an application server. This allows real thin clients because the GUI itself does not require high computation power. Additionally changes on the application logic can be done transparently on the servers without touching the clients.

- Multi-tier architectures
  The multi-tier architectures split the application layer of the 3-tier architecture into several layers itself and weaken the strict encapsulation of the business logic (see figure 6.36). On the one hand this weakens the architecture concept. On the other hand it supports the convenience of the user since the data is not transferred to the servers for every little action. The intention is to perform the small checks inside a small business layer on the client and keep the intensive calculations on the servers. The server side layer then contains the interface to the database as well.

CORBA based architecture The Common Object Request Broker Architecture (CORBA) has become a quasi-standard for client-server architecture within heterogeneous environments. This open standard has been defined by the Object Management Group (OMG) but the implementations of the broker-architectures are done by diverse companies. That is why a huge amount of different implementations are available, which are not always fully compatible with each other.

CORBA enables the communication within heterogeneous client and server environments where even different programming languages can be used to implement the different parts. This is achieved by the general definition of an Interface Definition Language (IDL). Each and every interface between client and server needs to be defined in this IDL. Using a special compiler these definitions can be translated into every target language where a client representation on the server side (server skeleton) and a server representation on the client (client stub) are created. These subjects act as proxies and handle, together with the Object Request Broker(ORB), all the communication between the client and server. The remote objects can be used as if they are locally available. For pure Java projects a similar but more comfortable concept called RMI can be used.

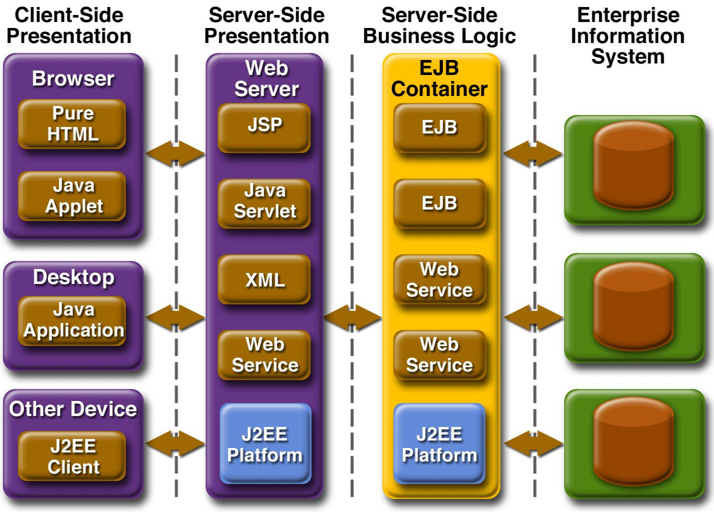


Figure 6.36.: General J2EE architecture (extracted from [Sun04a])

Java and RMI The concepts and structures of RMI are similar to CORBA but have the limitations of being developed only for Java. Therefore they can only communicate with Java components without additional adapters like IIOP (see below).

The programming language Java has been developed by Sun Microsystems and is a pure object-oriented and platform independent language. RMI is also able to talk to objects not implemented in Java using the Internet Inter-ORB Protocol (IIOP). This ensures that existing implementations can be incorporated into the framework developed in Java without a major rewrite. As an additional advantage the RMI methodology is an integral part of the J2EE[4] Environment and does not require an IDL, which saves the double effort of the interface definitions.

Client distribution

The PDTS system technology baseline had the advantage that it was not necessary to distribute any software client. All updates could be handled in the web server(s) directly and the users automatically use the latest version of the system. For the implementation of EVERIST a similar possibility was required. One objective of the PROMENADE project is that the software distribution of the presentation layer/client should be very easy and if possible without requiring client software installation. Additionally the efforts for updates and

[4] J2EE stands for Java2 Enterprise Edition which is a powerful development framework from SUN Microsystems for Java([Sun04a]). It supports the development of big projects with newest component based software development methodologies. On outline of the general J2EE architecture is presented in figure 6.36.

maintenance should be as low as possible. To achieve this the approaches outlined in the following subsections could be used.

Java applets Because the implementation of most of the clients was intended to be done in Java it would be possible to use Java applets as clients. These applets would be provided by a central Web-server. The user would download them and execute them inside a Web-browser. Because this approach would use the sandbox[5] principle it would be safe against misuse. Another advantage would be that the user downloads the latest applet and uses the newest version of the client every time. This eases the administration, which needs only to ensure that the web-server always carries the newest version. Therefore this approach has similar advantages like the PDTS web service approach. On the other hand the complete downloading of the applet every time implies additional network traffic and running as an applet requires the overhead of the browser framework itself. For those cases a stand-alone program with classical distribution has its advantages with the drawbacks of administration.

Java Web Start To combine the advantages described above without suffering from the disadvantages is the idea of the Java Web Start technology ([Sun04b]). Web Start is very easy to use and it does not need any extra development effort. Even legacy Java programs can be accessed via Web Start. The only prerequisite is that the client computer needs Web Start to be installed which comes automatically with the Java Runtime Environment (JRE). The idea of Web Start is presented in figure 6.37. Every time the application is started, the Web Start on the client checks for a newer version of the application JAR files on the server. If a newer version is available, it downloads and starts it, otherwise the locally cached application JARs are executed. This combines the central management and update with network traffic reduction and being a client application.

An additional advantage of Web Start is that it uses the sandbox principle, thereby limiting the access to resources to a pre-configured amount. Web Start provides a complete Java environment and if a different Java version is required Web Start allows installing that version in parallel to the existing version(s). Additional rights for an application can only be granted to certified applications ensuring that only authorized code is executed.

[5] The sandbox principle describes the hermetical encapsulation of the runtime environment. This implies that a running application only knows its own environment, his sandbox. Everything outside this box can not be seen nor accessed so that an applet only knows its browser environment.

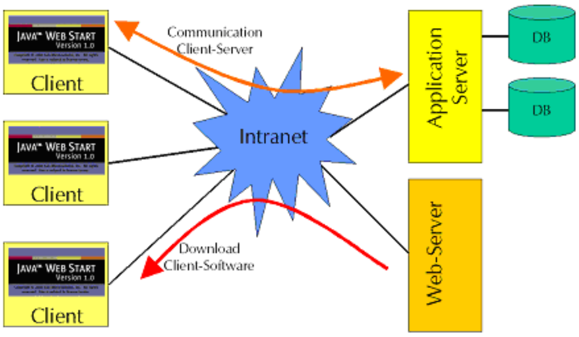

JAVA™ WEB START Version 1.0
Client
JAVA™ WEB START Version 1.0
Client
JAVA™ WEB START Version 1.0
Client
Communication Client-Server
Intranet
Download Client-Software
Application Server
DB
DB
Web-Server

Figure 6.37.: Java Web Start

Selection

Above the different State-of-the-Art architecture and software distribution techniques are described. In the beginning of the PROMENADE project the advantages and disadvantages of the different possible combinations of the architecture together with software distribution methods have been analyzed. The projects architecture group decided for a multi-tier architecture based on J2EE and the Java programming language together with the Java Web Start distribution method. For interfacing other legacy components like the simulation tools CORBA is used. Because the EVERIST system mainly needs to interface and interact with the PROMENADE Design Environment and this system is realized with the combination of the J2EE and Java Web Start technologies, the EVERIST system uses these technologies as well.

Resulting architecture

As outlined above, the EVERIST system uses a multi-tier J2EE architecture with Java Web Start distribution. The architecture overview of the system is depicted in figure 6.38. Each Tier and their modules will be described in the following sections.

Enterprise Information Systems The enterprise information systems provide means for the application server tier to store and retrieve data in/from a relational database system and a revision control system to handle file data.

The Relational Database Management System (RDBMS) role can be filled by any SQL92 compliant RDBMS, e.g. Oracle Database 10g or PostgreSQL, working together with the selected Application Server (see below). It is preferred to use Oracle 10g, or a newer version, because it is more scalable than other database management systems. The RDBMS provides an enterprise level relational database system where all primitive and non file based tracking data is reliably stored.

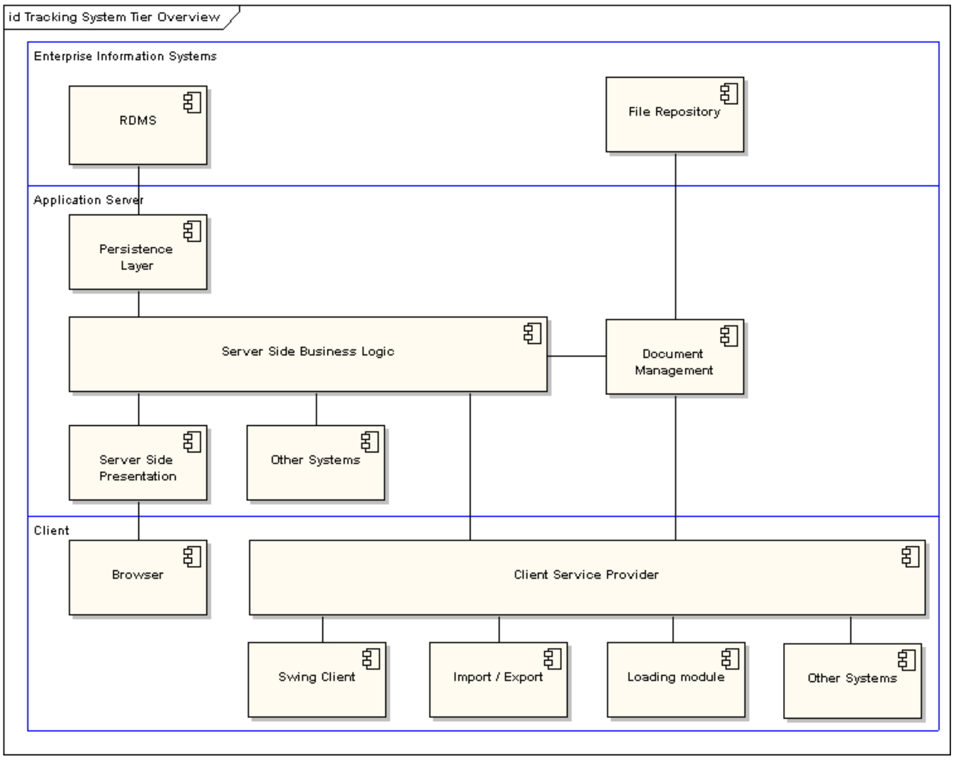


Figure 6.38.: Tracking System Tier Overview

For the realization of the Document Management component the revision control system Subversion is used. Subversion is a free and open-source version control system similar to the well known CVS. It offers a couple of advantages like easier application integration and better support for managing binary files (smaller differences). Revision control systems manage files and directories and their changes over time. A tree of files is placed into a central repository. The repository is much like an ordinary file server, except that it remembers every change ever made to the files and directories. It manages the files by registering only the differences between the different versions. This allows the user to recover older versions of the data, or examine the history of how the data changed.

Application Server The application server is a software platform that provides the services and infrastructure required to develop and deploy middle-tier applications. Middle-tier applications perform the business logic necessary to provide clients with access to enterprise information systems.

The preferred Application Server is JBoss but by staying very close to the J2EE specification, it should be possible to switch to other application server providers. The JBoss Application Server is the #1 most widely used Java application server on the market. JBOSS is a J2EE certified platform for developing and deploying enterprise Java applications, Web applications, and Portals. The JBoss Application Server provides the full range of J2EE 1.4 features as well as extended enterprise services including clustering, caching, and persistence.

Persistence Layer The persistence layer is a module that resides on the application server and provides all the logic for accessing the RDBMS. By using a separate layer to access the database, no database specific code has to be written. The technique used by the persistence layer is Container Managed Persistence(CMP) 2.0. By using CMP, the developers only have to specify the fields to persist and the framework will generate the necessary tables and logic to access the tables.

Server Side Business Logic The server side business logic module embodies business rules rather than viewing of data or storing of data. The Business Logic Layer(BLL) contains the logic to track data, and perform operations on the data. Because the tracking environment consists mainly of data management, all data management is done in the BLL. The BLL directly communicates with the Persistence Layer to actually perform the mutations in the database. All requests to the data have to be managed by the BLL. The BLL manages security, relations, locking and all the other logic that is shared by all the clients requiring to access the data.

Document Management The Document Management System (DMS) is a separate module which is loosely coupled to the BLL. This allows for possible exchanges of the DMS. The current DMS has an interface to the Subversion repository, described above, and the persistence layer. The persistence layer is used to save document meta information and the Subversion repository is used to save the file data. The DMS does not provide any means to structure the data. The structure is provided by the systems using the DMS.

Server Side Presentation The server side presentation is not yet implemented but is included in the diagram. This module is served by the Tomcat web server which is part of the JBoss Application Server. This module is foreseen to allow web clients next to the Java executable clients.

Service Facade The service facade does not provide any logic for the tracking environment but defines the application's boundary with a layer of services. These services establish a set of available operations and coordinates the application's response in each operation. In the figure it is shown as the "Client Service Provider".

Other Systems Other systems and not yet defined system components can either use the service facade or the business logic directly depending on the location of the component. If the other system is based on the same application server, it can use the business logic directly. If the system is located outside the application server, it can use the service facade or the services provided by the server side presentation layer.

Client Layer The client layer consists of several modules that can interact with the application server and are responsible for managing the user interactions. They exchange the data with the Application Server using the defined Data Transfer Objects(DTOs).

Flexible systems extensions To allow for flexibly extending the client functionality with additional modules, the architecture of the EVERIST system integrates a plug-in framework. This framework is based on the Open Services Gateway initiative (OSGi [OSG06b])[6] definitions. A modular application (as opposed to a monolithic one) having well-defined dependencies between the modules can be restructured into a plug-in application with OSGi. The OSGi framework itself consists only of interfaces and rules for which parties within or outside the alliance can write implementations. Several Java implementations of the framework exit, the EVERIST system uses the Equinox [OSG06a] implementation of the Eclipse project.

Because the EVERIST system is very modular, it is suitable to be used in combination with OSGi. Future plug-ins are likely to be used on the client side, therefore OSGi and its Eclipse Equinox implementation were adopted as a plug-in framework to be run solely on the client side. This allows modules to be added to the client side of the application in a very flexible way. Since dependencies go one-way, added modules can be considered optional. Whenever new functionality is added, considerations as to whether this functionality is part of the core system or should be implemented as a plug-in have to be made. Principally all functionality can be implemented as a plug-in. But if this requires the core system to disclose a large part of its functionality, it should be implemented within the core system. When the new functionality requires the core system to be dependent on it, it can not be implemented as a plug-in either and should be added to the core system. OSGi plug-ins are ideal for extending functionality as well as the user interface. Whenever new functionality is added to the EVERIST system as a plug-in, it most likely has to be integrated in the user interface. This is done using a registering pattern where plug-ins register themselves to the core system which shows them in the

[6] The OSGi alliance consists of a number of large companies. It was founded to provide a platform and device independent means to split up applications into parts and have these communicate among each other and the outside world through well-defined interfaces.

user interface. An example of a plug-in is the functionality for exporting and importing data from and into the system in XML format described in section 6.5.2.

6.5.5. Data models

This section should briefly highlight the data models used by the system. Roughly three different models can be distinguished, namely the domain data model, the Enterprise Java Beans model and the RDBMS data model. The domain data model is used to transfer data between the clients and the application server layer and is shown in figure 6.39. It represents the entities of the application domain, their inheritance model and the dependencies on support information like the histories of the different attributes. Apart from the clients, this Data Transfer Object (DTO) data model is and will be used by the Service Facade and diverse plug-in modules.

The other important data model is the one stored in the RDBMS. This model is shown in figure 6.40. As previously described this is automatically created by the persistence layer using the Container Managed Persistence (CMP) of the J2EE framework.

6.5.6. Embedding into environment

Besides the shared usage of the Document Management and the User Management components further integration of the different PROMENADE components is used. As outlined during the descriptions about the new entities Process Step&Flow in section 6.5.2, the PROMENADE Tracking component (EVERIST) is integrated with the other components. The integration with the Design component and the shared Parameter, Unit and Materials management component allows the instantiation of the recipes in the Tracking component from the Design component. This eases the tracking of the runs and unburdens the Design component from massive amounts of experimental data. Additionally the integration prevents double entry errors and relieves, together with the loading features, the process engineers from manually entering data. Overall the combination of the different components ensures reproducibility and process knowledge conservation.

An additional integration area is the usage of the Tracking component by the Simulation component to store simulation results. These results should be captured because they provide parts of the verification data and potentially require signification amounts of resources to be generated. Capturing them prevents unnecessary rerunning the simulations. The data is pushed into the Tracking component because the potentially massive amounts of data should not burden the Design component. The Tracking component has been designed specifically to handle big amounts of data.

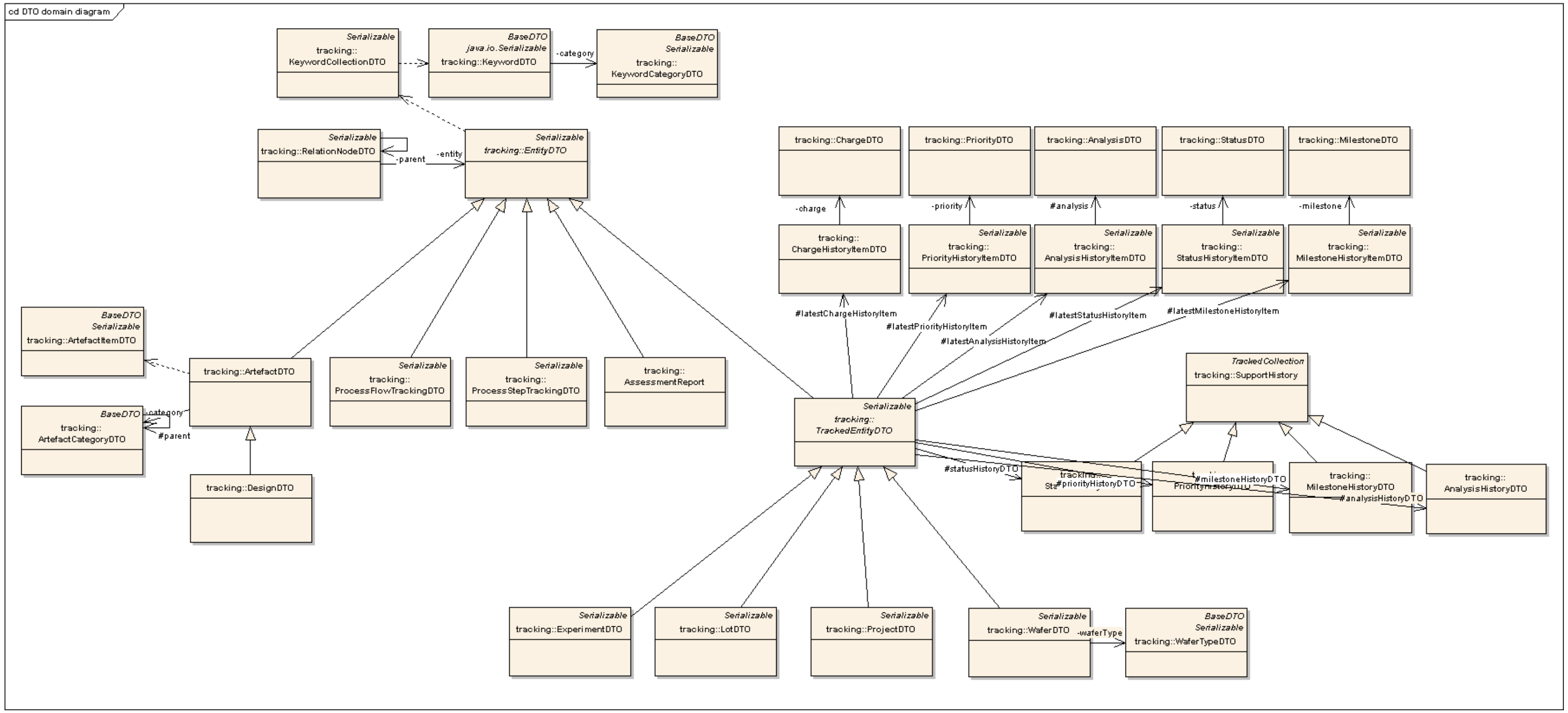


Figure 6.39.: Domain Object Model

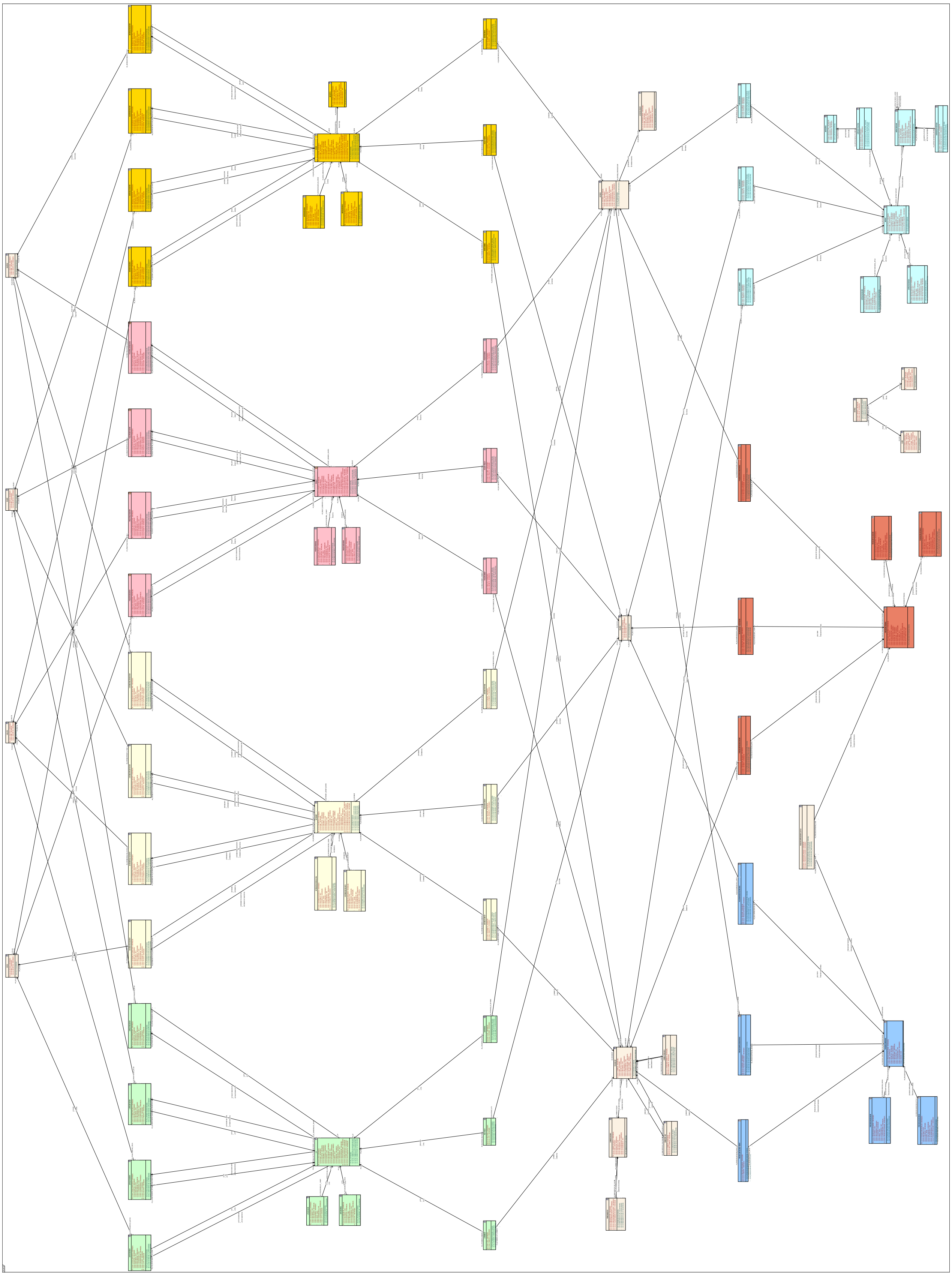


Figure 6.40.: Snapshot of the database data model of the EVERIST system

Although the systems can be integrated in the above outlined way, the Design and Tracking components can be used independent from each other. But they still rely on the basic components Document Management, User Management and Parameter, Units, Materials Management. This means that a commercial customer could opt to only use the Design or Tracking component. Anyhow the full power of the overall verification approach is only achieved by using the Design and the Tracking component in conjunction.

6.6. Back Annotation

The core task of the Back Annotation component is to shift data between different systems and environment components. Therefore it is responsible to interface with other systems to exchange and convert data between the PROMENADE components and external systems. This view is presented in figure 6.4. An additional envisioned task is to generate data, information and knowledge via analysing external or tracking data. Therefore it is the component closing the loop from the experimental verification back to the process design environment. Both areas are described in a bit more detail below although the current version of the Back Annotation modules are only a very rudimentary version. Much more ideas for this component are described in section 7.2 and are subject for future research and development.

6.6.1. Data loading

Apart from the relocation of the loading functionalities, the concepts for the data loading into the PROMENADE environment have been significantly changed as well. The loading into the PDTS system was implemented as a set of Perl scripts and modules implementing individual loading routines. They are to be used for all different loading jobs based on agreed conventions for directory locations and structures, file names and their contents, document structures etc. Therefore they were kind of hard coded for the initial implementation and the initial surrounding the system was used in. Within the PROMENADE Back Annotation component the concepts for the realization have changed to be user configurable, generic and being an additional plug-in component for the environment.

The user configurability will be managed in a way that several different modules for different loading purposes can be used. The user can define loading jobs where each job can be separately scheduled within a scheduler module executing the job at the times and recurrence the user has configured. The simplest job type takes a single directory and all files (not directories) in there can be loaded into one artefact (as individual documents) or into individual artefacts. For these the artefact category as well as the keywords to be set can be defined. The names, titles and descriptions are configurable and the user can use regular expressions to place parts of the file name into the different fields. Additionally a file mask can be set defining which part of the file name is used to identify related entities of definable entity type. This could e.g. be used to relate a Runcard with the name ”E1234.xls” automatically to the lot with the Name ”E1234” via setting an appropriate name. On top of that the user can set what should happen if the related entity can not be found. He has the choices to ignore this fact or to let the system create the related entity and relate it afterwards.

6.6.2. Knowledge generation concepts

Within the PROMENADE project only conceptual work for the knowledge generation has been planned. This is due to the fact that without a stable working baseline system no real implementations of div. back annotation ideas were possible. Therefore the further descriptions only highlight a very initial conceptual step and defer further conceptions of other ideas described in section 7.2 and prototype implementations to future research.

First conceptual work has been spent into the idea to try to (semi-) automatically generate rules for the Design environment from the Tracking component data. This rule generation would allow to perform more and better Design For Manufacturabilty checks. Therefore the Design environment could use more abstract knowledge to assess manufacturing sequences. By this it would be possible to improve the quality of the process blue book used for the first experimental verifications for a new device manufacturing process.

As a first step to evaluate the feasibility of this idea, a literature study about existing applicable techniques has been performed. Here different approaches like OLAP, Warehousing, Artificial Intelligence, Sematic Web, Ontologies and Data Mining have been reviewed. Because the graph representation used in the current system can be extended relatively easily into a semantic network, as descried in section 7.1.1, this representation together with Artificial Intelligence approaches seems to be a promising route forward. An alternative potentially useful approach is the usage of Data Mining techniques, especially the clustered data mining. Both pathes and potentially some others will be evaluated further in the remaining time of the PROMENADE project.

6.7. Conclusions

This chapter presented a novel approach for thin film, silicon based MEMS process development support. Based on the idea of a concurrent structural and process design the necessity for a Design For Manufacturability approach has been discovered. The process DFM can be covered by three phases of process verification namely process flow consistency checking, process flow simulation and support for experimental verification. While the first two verification steps are covered by other components of the PROMENADE environment, the focus for this thesis was set on the support of the experimental verification. For that area a new approach has been developed which has been implemented into two different systems. For that concept a patent application has been filed which is published under [Ort05b]. The first prototype, called PDTS, was designed as a requirements evaluation vehicle using technologies appropriate for this purposes. The second implementation of the structural approach is embedded into the PROMENADE environment and uses an improved, enterprise suitable architecture and technologies. This implementation contains additional concepts and features improving the structural approach. The basic concepts and both implementations have been discussed. The newly developed concept additions making the whole approach more versatile have been introduced as well. One of these new concepts addresses the industry shortcoming of having no technical possibility to electronically transfer process IP. This has been tackled by the introduction of the selective export and import features. Finally some promising ideas for knowledge generation making the PROMENADE system a self-learning environment have been outlined.

7. Further Work

The current chapters briefly summarises ideas for future work or extensions of the current systems and concepts which could not be realized within the time frame of this thesis.

7.1. Useful extensions of the tracking environment

During the efforts for the conception and development of the current EVERIST system the following extensions of the concepts and functionalities were discovered to be useful.

7.1.1. Directed & attributed relations

Analyzing the problem domain it can be noted that the relations between the different items represent a major part of the information and knowledge. Therefore the relations need to be explicitly modelled to extend the expressiveness. This can be achieved by a feature allowing the relations to be categorized into configurable relation types via additional attributes. The idea behind that is: If items belong to each other this is information. If wafers got similar processing, if assessment results are similar, etc. they should become related to each other and via a certain relation type the different kinds of similarities can be expressed. In saying these items are related because they are similar concerning one respect (categorization type) e.g. (similar process flow, similar experiment scope, etc.) this knowledge can be specified in all of its diversity. Because the relation categories are not always bi-directional, an additional concept extension substitutes the bi-directional or undirected relations with uni-directional ones. For bi-directional attributes this can be modelled with one relation for each direction. Both changes together convert the undirected graph into a directed, attributed graph, a semantic network. The knowledge captured this way can be used for later data analysis and potentially for the generation of new knowledge. The major advantage of these extensions are that this can be used to potentially generate new rules for designing processes making the Verification of Manufacturabilty step more powerful and enabling a learning and reasoning system via Artificial Intelligence approaches. Therefore it lays a baseline for the future extensions of the Back Annotation component described below.

7.1.2. Navigation by Relation Types

The addition of the relation types would open another possibility to navigate the information graph. By selecting a set of relation types, a collection of one or more types, the amount of entities shown in the Relation Explorer and in the Relation Graphs could be reduced to a better manageable amount. These views then would show only those entities related to an already presented entity via a relation having a type of the defined set. This functionality would allow inspecting the information network for only a reduced set of aspects the user might be interested in. The usability and the specific retrieval possibilities would be improved by this. In conjunction with the selective export functionalities this offers additional flexibility.

7.1.3. Relation groups

Another new useful concept would be the addition of relation groups. They would allow to group certain relations, potentially but not necessarily by the relation type, together to enable the capturing of related relations. These could be useful to e.g. group all relations of one project together. These groups could be implemented into the Relationship Management dialogue and increase the expressiveness of the relations.

7.1.4. FMEA integration

To integrate a FMEA mechanism into the Tracking Environment would be another useful addition to the system. This would allow to easily deduct such an analysis from the process steps and flows to prevent work intensive replication of this data in a spreadsheet. To allow this, only the necessary parameters need to be added to the steps and flows and a spreadsheet integration could analyse the most severe factors.

7.1.5. Additional retrieval mechanisms

Apart from the existing retrieval and navigation mechanisms the following additions are foreseen to be realized shortly:

Lot status report The lot status report is intended to give an overview about the current lot progress of all unfinished lots in the manufacturing line. For this report the status, milestone and priority fields of the lots should be used in conjunction with the start date and planned end date. Presenting this data in a concise report enables an automatic overview about all activities. A traffic light system should enable the easy identification of deviations.

Change report The idea for the change report is to be able to view a report of all changed item within a configurable time span. This report is based on the already build in mechanisms to track all changes in the system. This allows to easily extract all activities during the given time interval. A special application of this report would be to present this change report automatically when a user logs in showing all changed items since his last login. This would give a more active system to the users.

Generic report generator Another useful addition will be the integration of a generic report generator. This module will enable the user to configure his own reports to extract certain aspects of the data. For the realization flexible report components like Crystal Reports could be used.

Searching Another area of extension is the creation of more flexible and powerful searching mechanisms. Via extending the expressiveness of the filters to be able to define Boolean functions over the complete database, more detailed but also more complex searches could be performed.

Full text searching The current implementation of EVERIST supports detailed filtering. But it is not possible to perform a full text search within text based artefacts e.g. MS Word or Excel documents, PDF documents, etc. Via using and interfacing an indexing server this functionality could be added to the system to allow for this type of searching. The user than would have the possibility to do text searching only in the database, only in documents or in both. This feature addition would make the detailed searching for certain strings much more powerful.

7.1.6. Further usability improvements

- Implement relation drag and drop
  The implementation of drag and drop functionality for relations could ease the relation management. This could be realized by just dragging and dropping on entity to another and by this creating a new relation.

- Implement relation management by drawing in graph
  Another possibility to manage relationships would be to allow drawing of relationships in the Relation Graph.

- Internationalization
  In the moment the system uses English language. Currently efforts are ongoing to externalize all strings and to allow configurable languages via language string files.

- Implement WebDAV interface
  A WebDAV interface could be added to the system to more easily man-

age the documents. Via such an interface the documents could be integrated into a normal file explorer and additions and retrieval could be handled without using the regular interface. The interface would handle the IPR protection and security management in accordance to the graphical user interface and update the corresponding document storage in the system.

7.2. Back Annotation

Another future development path is the realisation and extension of the Back Annotation component of the Process Design and Tracking Environment. Usable techniques could be (not limiting):

- Usage of artificial intelligence tools to retrieve rules and dependencies from the data pool in the tracking environment.
- Usage of warehouse methods and data mining tools to correlate data and extract rules and dependencies from the data pool in the tracking environment.
- Try to apply different Business Intelligence approaches for retrieving information and knowledge from the EVERIST data pool.
- Integration of graphical analysis packages to (semi-)automatically extract geometries from SEM images.
- (Semi-)automatically re-adjust the process step and flow parameterizations in the design environment from data collected in the tracking environment (deviations from planned result parameters).
- (Semi-)automatically feed the design environment with result parameters from the assessments in the tracking environment.
- Interface with MES systems like PROMIS to collect more detailed fabrication data and allow for easier status update.
- Interface with commercial software, e.g. ERP systems to allow for cost tracking and planning.

7.3. Integration of Structural and Behavioral design

The next necessary development step in the support of silicon based MEMS process design is the integration of the concurrent structural, behavioural and process design approaches. They need to be combined into a microsystem process and product design approach based on knowledge. The goal of these

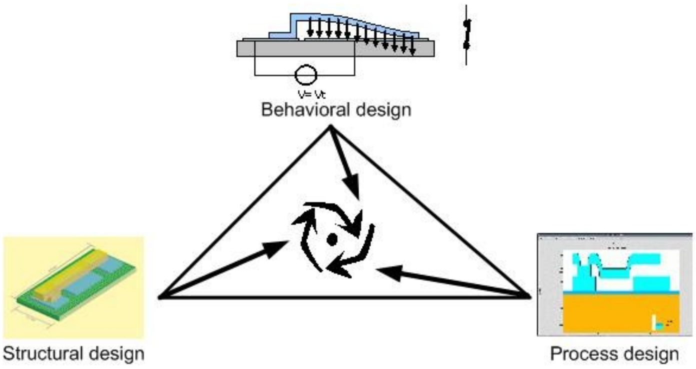


Figure 7.1.: Design approach triangle

activities would be to develop an overall concurrent design model covering and integrating the aspects of structural and behavioural design approaches on the one hand and the structural and process design tasks on the other hand. The interdependencies between the top-down (from behaviour and structure) or bottom-up (from technology and process flow to structure) require a concurrent approach. Such work could start with efforts to define a concurrent design model integrating all three necessary aspects. As a foundation for this work the state of the art status concerning design strategies, flows and environments would need to be evaluated and summarised.

The first task of such future developments has to try to assess approaches supporting the different angles/dimensions of the MEMS design and identify potential candidate environments covering the angles. These different angles are depicted in the design triangle of figure 7.1. This diagram tries to present the different aspects essential for MEMS design:

- The design of the dynamic behaviour of the intended device. This modelling is typically done in finite element modelling software packages like CoventorWare from Coventor [Cov05] or IntelliSuite 8 from IntelliSense [Int05a].

- For the design of the structural aspects of a device currently mostly tools from the electronic design areas are used. Tools like CoventorWare and IntelliSuite8 can be used but typically only for single devices while a complete microsystem mostly requires a few devices up to several thousand of devices.

- For the process design only very few tools are available on the market. For this task the tools of the FP6 project PROMENADE [Eur04] or partly the tools of PhoenixBV [Pho05] can be used. For simulating the processes the typical process simulation environments like Silvacos [SIL05b] Athena (a module of PROMENADE as well) can be used.

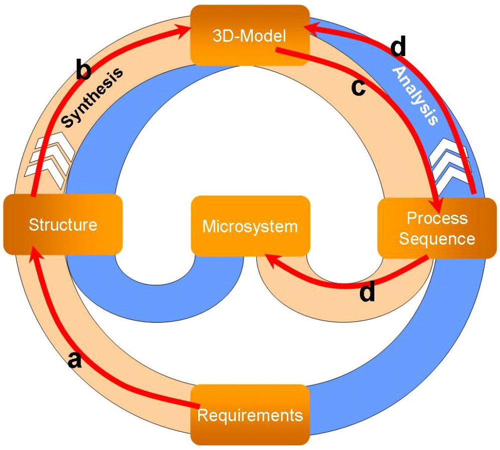


Figure 7.2.: Pretzel model for concurrent structural and process design

The second task of such activities would need to look into developing a unified design methodology covering all three angles which can be supported by the potential candidate environments. As an example, up to now a concurrent design method for the structural and process dimensions has been developed and published. This so called pretzel model is presented in figure 7.2 and is used as a basis for the PROMENADE Design Environment.

This design approach is designed especially for the implementation in a powerful software environment like demonstrated within the EU PROMENADE project. A possible design flow for a complete new device with unknown process flow could be as laid out in figure 7.2. The model illustrates the possible flow: The designer starts with creating a structural description based on the requirements (step a). A corresponding 3D-Model of the device (step b) can be derived from the structural description. Now a process sequence for manufacturing the device can be designed (step c). Additionally important material and process step data is collected and passed to the knowledge base. The process sequence can now be verified by simulation based on the data in the knowledge base (step d). Step c and d have to be repeated until the process sequence matches the 3D-Model. With the growing knowledge base the concurrent design becomes more and more straight forward. This model can be taken as a basis for integration of the concurrent behavioural design.

The third task would be to identify and define interfaces on an abstract level between the different selected design environments. For that the design methodology has to be projected onto the selected systems and the necessary interfaces have to be identified to support a round-trip design flow. This step is to perform a pre-feasibility study to evaluate whether the developed methodology can be supported by a system consisting of integrated existing design and management tools.

8. Conclusions

As outlined in section 2.4 the work for this thesis was supposed to address the following topics:

Product Engineering Method As identified in the State-of-the-Art discussion, no Product Engineering method for silicon based MEMS process and design IP could be found in literature. Therefore one aim of the work was to define such a development method. For that the well established product development method Stage-Gate™ was combined with the European standard project management method PRINCE2. The combination gave a good starting point for a method suited especially to the needs in the targeted industry. A newly defined process development method has been fitted into this baseline to specifically enable efficient and target focussed process development efforts. The detailed development sequence with all necessary deliverables has been described and the management of the efforts have been woven into the development process. A framework of management tools and components have been defined as well. Altogether the developed method and components add significantly to the development practice and fulfil the needs of the business environment.

Product Engineering Framework The new development methods requires a framework of tools, components and processes to support the execution. Because no suitable framework for this industry was preexisting, such a framework has been created. It is based on various standard software components (commercial and where possible Open Source) supporting the method in all necessary areas. Additionally novel tools have been developed, integrating the different standard components to form a coherent framework for the process and design IP developments. The method and the framework has been introduced in a company and is used to guide their developments.

Process Development support The third major issue addressed with this work are the reproducibility and knowledge management issues detected in process development efforts. As motivated in the State-of-the-Art descriptions, the encountered development practice did not use and did not have software tools sufficiently supporting the experimental verification efforts. This has been addressed by the development of the novel structural approach which has been implemented into the PDTS requirements evaluation prototype and into the PROMENADE Tracking Environment (EVERIST) system. The EVERIST system is able to collect all data of experimental verification efforts, is able to group them together and allows the information retrieval by several different means. Together with the other PROMENADE components the environment builds a central working vehicle for the process engineers to design, check, simulate and track the development efforts. Therefore the PROMENADE Tracking environment becomes a central knowledge base for all experimental verification data and can be used for all sorts of information retrieval.

Process IP transfer For the issue of non-existing electronic process IP transfer mechanisms a new approach has been invented and implemented into the EVERIST system. It supports the experimental verification tasks in an optimal way allowing to collect all data, information and knowledge. The build in export mechanisms allow for a very fine grained selective export of the collected information and therefore enables to transfer certain process IP packages between systems. The export is stored into an open, XML based format so that it can be imported into other instances of the EVERIST system or into any other system having an appropriate import filter. The XML schema definition is published on a web site and can be used by all tools interested in using exported data.

The current state of the EVERIST system can be considered to be a solid baseline for future extensions outlined in section 7. The systems will be extended further in upcoming new research projects with the above concepts to improve the expressiveness and retrievability of the collected data and information. Abstract knowledge generation in the form of rules for the first step of the verification sequence is an interesting further research topic which hopefully will be addressed in future EU research projects. A commercialization of the EVERIST system is foreseen for the upcoming months.

List of Figures

List of Tables

9. Bibliography

[Adv05a] AdvanTech. *ActionBase*. http://www.actionbase.com/products/main.asp. June 2005

[Adv05b] Advisor.ca. *Vision: The prerequisite to building a better team*. http://www.advisor.ca/practice/special_report/article.jsp?content=21671. August 2005

[Ali05] Ali.as Software. *CVSmonitor*. http://ali.as/devel/cvsmonitor/. July 2005

[All05] All About MEMS. *MEMS FAQ*. http://www.allaboutmems.com/memsfaqs.html. September 2005

[And96] Anderson, Richard E. *Phased product development: friend or foe?* http://www.findarticles.com/p/articles/mi_m1038/is_n6_v39/ai_19125123. November 1996

[App05a] Applied Materials. *Applied FAB300*. http://www.appliedmaterials.com/products/fab_300.html?menuID=1_7_1. September 2005

[App05b] Applied Materials. *Applied WorkStream*. http://www.appliedmaterials.com/products/workstream.html?menuID=1_7_2. September 2005

[ASE95] ASEE/IEEE Frontiers in Education. *The ECSEL's Integrated Approach to Industry-Academe Relations*. http://fie.engrng.pitt.edu/fie95/2a1/2a13/2a13.htm. October 1995

[Aud06a] Auditnet.org. *Auditors Sharing Audit Programs*. http://www.auditnet.org/freeap.htm. January 2006

[Aud06b] Auditnet.org. *Project Management Audit Procedures*. http://www.auditnet.org/docs/Project%20management%20Audit%20Program.doc. January 2006

[Ben02] Bentley, Colin: *PRINCE2 a practical handbook*. 2nd edition. ComputerWeekly, 2002. – ISBN 0–7506–5330–0

[Ber01] BERCZUK, Steve: Teamwork and Configuration Management: Some Patterns http://www.berczuk.com/pubs/CMPatterns_C++Report.pdf. (2001)

[BOI05] BOIN GMBH. *Wafermap*. http://www.boin-gmbh.com/. August 2005

[Bri03] BRITISH STANDARD INSTITUTE. *ISO 10006:2003: Quality management systems – Guidelines for quality management in projects*. July 2003

[Bro98] BROOKS AUTOMATION, INC. *Domain Expands and Elevates Starfire Capabilities with Release of Starfire 3.0*. http://investor.brooks.com/releasedetail.cfm?ReleaseID=68461. December 1998

[Bro05] BROOKS AUTOMATION, INC. *PROMIS 2 - Configurable Manufacturing Execution System*. http://www.brooks.com/pages/215_promis.cfm. June 2005

[Bru05] BRUNS, IRIS AND REUTH, RALPH. *Innovation als image- und bilanzwirksamer Erfolgsfaktor im demographischen Wandel*. http://www.iaw.rwth-aachen.de/download/publikationen/udz1_2005_bruns.pdf. March 2005

[Buc05] BUCHANAN, J.R. *The GDSII Stream Format*. http://www.buchanan1.net/stream_description.shtml. September 2005

[Bug05] BUGZILLA.ORG. *Bugzilla - Issue Tracking System*. http://www.bugzilla.org. August 2005

[Bux05] BUXTON DESIGN. *Performance by Design: The Role of Design in Software Product Development*. http://www.cs.ubc.ca/~ksbooth/buxton.pdf. August 2005

[Car05] CARNEGIE MELLON SOFTWARE ENGINEERING INSTITUTE. *CMMI Homepage*. http://www.sei.cmu.edu/cmmi/. August 2005

[CKS03] CHRISSIS, Mary B. ; KONRAD, Mike ; SHRUM, Sandy: *CMMI: Guidelines for Process Integration and Product Improvement*. 1st edition. Addison Wesley Professional, 2003. – ISBN 0321154967

[Coc96] COCKBURN, A: The Interaction of Social Issues and Software Architecture, http://alistair.cockburn.us/crystal/articles/isiasa/ACMinteractionofsocissswarch.pdf. (1996)

[COL05] COLLEGE HOUSE ENTERPRISES, LLC . *Product Engineering and Manufacturing*. http://www.collegehousebooks.com/Product%20Engineering%20and%20Manufacturing.htm. September 2005

[Com04] COMPUTERWORLD. *Quality Model Mania*. http://www.computerworld.com/developmenttopics/development/story/0,10801,90797,00.html. March 2004

[Coo98] COOPER, Robert G.: *Product leadership – Creating and Launching Superior New Products*. 1st edition. Basic Books, 1998. – ISBN 0–7382–0156–1

[Cop94] COPLIEN, J.O.: A Generative Development Process Pattern Language ftp://st.cs.uiuc.edu/pub/patterns/plop-papers/ProcessPatterns.ps. (1994)

[Cov05] COVENTOR, INC. *Coventor Home Page*. http://www.coventor.com/. September 2005

[Dav04] DAVIDSEN, BJOERN ARE. *Innovation and product development - methods and tools*. http://www.telenor.com/telektronikk/volumes/pdf/2.2004/Page_080-081.pdf. March 2004

[DBG+03] DESCAMPS, Gilles-Eric ; BAGALKOTKAR, Satish ; GANESAN, Subramanian ; IYENGAR, Satish ; PIRSON, Alain: Design of a 17-million gate network processor using a design factory. In: *DAC '03: Proceedings of the 40th conference on Design automation*. New York, NY, USA : ACM Press, 2003. – ISBN 1–58113–688–9, S. 844–849

[Dep96] DEPARTMENT OF DEFENSE. *DoD Guide to Integrated Product and Process Development*. http://www.acquisition.gov/comp/seven_steps/library/dod-guide-to-integrated.pdf. June 1996

[Des05] DESIGN AND REUSE. *The Catalyst of Collaborative IP Based SoC Design*. http://www.us.design-reuse.com/. June 2005

[Doc05] DOCUMENTUM. *Enterprise Content Management*. http://www.documentum.com. 2005

[Dol05] DOLPHIN INTEGRATION. *Silicon IP*. http://www.dolphin.fr/flip/flip_siptovic.html. June 2005

[DRM05a] DRM ASSOCIATES. *Failure Modes and Effects Analysis (FMEA)*. http://www.npd-solutions.com/fmea.html. August 2005

[DRM05b] DRM ASSOCIATES. *PD-TRAK® NPD Process Tools*. http://www.npd-solutions.com/pdtrak.html. August 2005

[Eed05] EEDO. *Eedo ForceTen*. http://www.eedo.com/products/forceten.html. June 2005 [eGr05] EGROUPWARE.ORG. *eGroupWare - Enterprise Collaboration*. http://www.egroupware.org/. August 2005

[Ell05] ELLEMTELL. *Programming in C++, Rules and Recommendations*. http://www.doc.ic.ac.uk/lab/cplus/c++.rules/. August 2005

[Eur04] EUROPEAN COMMISSION. *PROMENADE - Process management and design system for microsystem technologies*. Specific targeted research or innovation project, Contract 507965, IST-2002-2.3.1.2, http://dbs.cordis.lu/fep-cgi/srchidadb?ACTION=D&SESSION=207032005-4-27&DOC=2&TBL=EN_PROJ&RCN=EP_CT_D:Specific=FP6_PROJ. 2004

[Eur05] EUROPEAN FEDERATION OF CHEMICAL ENGINEERING. *EFCE Bologna Recommendations*. http://www.efce.info/Bologna_Recommendations-design-.html. September 2005

[Exc05] EXCELERATE TECHNOLOGIES, LLC. *ProcessNavigator for Inspection*. http://www.exceleratetech.net/pn_dva_inspection.asp. August 2005

[Exp05] EXPERIENCE IN SOFTWARE. *Project Kickstart*. http://www.projectkickstart.com/html/pkswin3.htm. August 2005

[Fab05] FABLESS SEMICONDUCTOR ASSOCIATION (FSA). *Understanding the Semiconductor Intellectual Property (SIP) Business Process*. http://www.us.design-reuse.com/articles/article6507.html. September 2005

[Fre05a] FREE SOFTWARE FOUNDATION. *GNU Bash*. http://www.gnu.org/software/bash/bash.html. July 2005

[Fre05b] FREE SOFTWARE FOUNDATION. *GNU CVS*. https://www.cvshome.org/. July 2005

[Fre05c] FREE SOFTWARE FOUNDATION. *GNU Make*. http://www.gnu.org/software/make/. August 2005

[Fre05d] FREE SOFTWARE FOUNDATION. *GNU RCS*. http://www.gnu.org/software/rcs/rcs.html. July 2005

[Fre05e] FREEBSD. *CVSweb*. http://www.freebsd.org/projects/cvsweb.html. July 2005

[GC02] OF GOVERNMENT COMMERCE, Office: *Managing Successful Projects with PRINCE2*. 3rd edition. OGC, 2002. – ISBN 0–11–330891–4

[Ger05] GERALD KOTONYA, IAN SOMMERVILLE. *Requirements Engineering - An introduction*. http://www.comp.lancs.ac.uk/computing/resources/re/slides/Ch1intro.ppt. August 2005

[Gid85] GIDO, JACK: *An Introduction to PROJECT PLANNING*. Industrial Press, 1985. – ISBN (0–8311–) 1160–7

[GK98] GERALD KOTONYA, Ian S.: *Requirements Engineering - Processes and Techniques*. 1st edition. John Wiley & Sons, 1998. – ISBN 0471972088

[Glo05] GLOBODIGITAL.NET. *Mambo - SupportCenter*. http://www.globodigital.net/Products/Support_Center/1/1/. August 2005

[Gro06] GROUP, Praxiom R. *ISO 9000 AN INTRODUCTION*. http://praxiom.com/iso-intro.htm. January 2006

[HWPB03] HAHN, K. ; WAGENER, A. ; POPP, J. ; BRÜCK, R.: Process Management and Design for MEMS and Microelectronics Technologies. In: *Proceedings of SPIE: International Symposium on Microelectronics, MEMS, and Nanotechnology 2003, Perth*, 2003. – SPIE Perth 2003

[IBM05a] IBM. *CATIA V5*. http://www-306.ibm.com/software/applications/plm/catiav5/. June 2005

[IBM05b] IBM COOPERATION. *Lotus Notes*. http://www.lotus.com/products/product4.nsf/wdocs/noteshomepage. June 2005

[IBM05c] IBM (RATIONAL). *Rational ClearCase*. http://www-306.ibm.com/software/awdtools/clearcase/. July 2005

[IBM05d] IBM RATIONAL. *Rational RequisitePro*. http://www-306.ibm.com/software/awdtools/reqpro/. August 2005

[IBM05e] IBM RATIONAL EDGE. *Project Planning Best practices*. http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/aug03/m_projectplanning_dd_ec.pdf. August 2005

[Inc05] INC., Synopsys. *Synopsis Website*. www.synopsis.com. August 2005

[Ins04] INSTITUTE, Project M.: *A Guide to the Project Management Body of Knowledge*. 3rd edition. Project Management Institute, 2004. – ISBN 193069945X

[Ins05] INSTITUT FÜR MIKROSYSTEMTECHNIK, CAVENDISH KINETICS B.V.: *PROMENADE - Manual Version 1.2*. Universität Siegen, Cavendish Kinetics B.V., December 2005

[Int05a] INTELLISENSE, CORP. *IntelliSense Home Page*. http://www.intellisensesoftware.com/index.html. September 2005

[Int05b] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 10006: Quality management systems – Guidelines for quality management in projects*. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=36643&ICS1=3&ICS2=120&ICS3=10. September 2005

[Jan05] JANHAGER, Jenny: *User Consideration in Early Stages of Product Development - Theories and Methods*, KTH Machine Design, PhD thesis, 2005

[Kar94] KARBHARI, V. M. *Design Methodologies*. http://www.wtec.org/loyola/polymers/c7_s3.htm. April 1994

[Lö05] LÖVGREN, Rolf. *Product development methods: Leonardo Da Vinci versus modern engineering education*. http://www.idp.mdh.se/personal/rln01/Produktutveckling/Leonardo%20som%20produktutvecklare%20(eng).pdf. November 2005

[LO06] LANGENHUISEN, S. ; ORTLOFF, D.: An Approach to Generate Knowledge to Support Silicon Based MEMS Development. In: *Proceedings of ICME: 5th CIRP International Seminar on Intelligent Computing in Manufacturing Engineering, Ischia*, 2006. – CIRP ICME'06. – ISBN 88–95028–01–5, 978–88–95028–01–9

[Lud05] LUDWIG CONSULTING SERVICES, LLC. *Requirements Management Tools*. http://www.jiludwig.com/Requirements_Management_Tools.html. August 2005

[Mar05] MAR, WILSON. *Project Planning Strategies and Tools*. http://www.wilsonmar.com/1projs.htm. August 2005

[Met05] METAQUEST SOFTWARE INC. *Census Issue Tracking*. http://www.metaquest.com/Solutions/BugTracking/BugTracking.html?source=google&keyword=issuetracking. August 2005

[MG05] MENTOR GRAPHICS, Cooperation. *Mentor Graphics Website*. http://www.mentor.com. August 2005

[Mic05a] MICRO AND NANOTECHNOLOGY COMMERCIALIZATION AND EDUCATION FOUNDATION (MANCEF). *Micro and Nanotechnology Commercialization and Education Foundation*. http://www.mancef.org. September 2005

[Mic05b] MICROSOFT COOPERATION. *Microsoft Content Management Server*. http://www.microsoft.com/cmserver/. July 2005

[Mic05c] MICROSOFT COOPERATION. *Microsoft Netmeeting*. http://www.microsoft.com/windows/netmeeting/. July 2005

[Mic05d] MICROSOFT COOPERATION. *Microsoft Remote Desktop*. http://www.microsoft.com/windowsxp/using/mobility/getstarted/remoteintro.mspx. July 2005

[Mic05e] MICROSOFT COOPERATION. *MS Exchange*. http://www.microsoft.com/exchange/default.mspx. June 2005

[Mic05f] MICROSOFT COOPERATION. *MS Project*. http://office.microsoft.com/en-us/FX010857951033.aspx. June 2005

[Min05] MINITAB INC. *MINITAB Statistical Software*. http://www.minitab.com/products/minitab/default.aspx. August 2005

[MJI05] MJI TEAMWORKS. *MJI Teamworks*. http://www.mjiteamworks.com/Site/index.asp. August 2005

[MRI05] MRI MONTGOMERY RESEARCH. *A Fabless Model for the Development of MEMS-Enabled Products*. http://www.future-fab.com/documents.asp?d_ID=1228#. June 2005

[MyS05a] MYSQL AB. *Eventum Issue/Bug Tracking System*. http://dev.mysql.com/downloads/other/eventum/. August 2005

[MyS05b] MYSQL AB. *MySQL database server*. http://www.mysql.com/products/mysql/. July 2005

[Nat67] NATHANSON, H.C; et a.: The Resonant Gate Transistor. In: *IEEE Trans. Electron Devices Volumn 14*, 1967. – SensorExpo2002

[NE00] NUSEIBEH, B. ; EASTERBROOK, S.: Requirements Engineering: A Roadmap, http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf. (2000)

[NIS05] NIST - NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Product Engineering*. http://www.mel.nist.gov/msid/pe.htm. August 2005

[NSB05] NSBA. *Creating a Vision*. http://www.nsba.org/sbot/toolkit/cav.html. August 2005

[Off05] OFFICE OF GOVERNMENT COMMERCE. *IT Infrastruture Library*. http://www.ogc.gov.uk/index.asp?id=2261. June 2005

[Ols04] OLSON, ERIC L. AND BAKKE, GEIR. *A comparison of new product development practices in North America and Norway*. http://www.telenor.com/telektronikk/volumes/pdf/2.2004/Page_120-125.pdf. June 2004

[Ope05a] Open Source. *Mambo Content Management Server*. http://www.mamboserver.com/. July 2005

[Ope05b] Open Source. *PuTTY*. http://www.chiark.greenend.org.uk/~sgtatham/putty/. July 2005

[ope06] openPR. *Was fehlt ist Traceability*. http://openpr.de/news/58341.html. January 2006

[Ora05a] Oracle Cooperation. *Oracle Collaboration Suite*. http://www.oracle.com/collabsuite/index.html. June 2005

[Ora05b] Oracle Cooperation. *Oracle Files*. http://www.oracle.com/collabsuite/feature_files.html. June 2005

[Ora05c] Oracle Cooperation. *Oracle(Peoplesoft) CRM*. http://www.oracle.com/applications/peoplesoft/crm/ent/index.html. June 2005

[Ort05a] Ortloff, Dirk: A Product Engineering Framework for MEMS IP Developments. In: *Proceedings of the 10th International Conference on the Commercialization of Micro and Nano Systems, Baden-Baden*, 2005. – COMS 2005

[Ort05b] Ortloff, Dirk. *INFORMATION MANAGEMENT AND TRACKING SYSTEM (IMTS)*. http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2005093517&F=0&QPN=WO2005093517. October 2005

[Ort06] Ortloff, D.: A Product Engineering Method for MEMS IP Developments. In: *Proceedings of ICME: 5th CIRP International Seminar on Intelligent Computing in Manufacturing Engineering, Ischia*, 2006. – CIRP ICME'06. – ISBN 88–95028–01–5, 978–88–95028–01–9

[OSG06a] OSGi Alliance. *equinox – an OSGi framework and a whole lot more* . http://www.eclipse.org/equinox/. January 2006

[OSG06b] OSGi Alliance. *The OSGi Service Platform - Dynamic services for networked devices*. http://www.osgi.org/. January 2006

[OVC05] Ortloff, D. ; Veenstra, B. ; Cooijmans, F.: A Systematic Approach Towards Reproducibility and Tracking of MEMS Process Development. In: *Proceedings of the 10th International Conference on the Commercialization of Micro and Nano Systems, Baden-Baden*, 2005. – COMS 2005

[Par05] PARTNERS FOR THE ADVANCEMENT OF COLLABORATIVE ENGINEERING EDUCATION (PACE). *Automotive life cycle management*. http://www.pacepartners.org/index.html. September 2005

[Pho05] PHOENIX BV. *Micro Mechanics/MEMS - Processing software*. http://www.phoenixbv.com/subpage.php?pid=51500. June 2005

[PMF05] PMFORUM. *Organizational Project Management Maturity Models*. http://www.pmforum.org/standards/matmatrix.htm. September 2005

[PMI05] PMI. *Introdution to Project management*. http://www.ise.canberra.edu.au/u6678/L-Schw-4ed/ch01.ppt. August 2005

[Pop05] POPP, Jens: *Ein Konzept für die Datenhaltung in der fertigungsnahen Mikrosystemtechnik*, Universität Siegen, PhD thesis, 2005

[Pro05a] PRODUCT DEVELOPMENT INSTITUTE. *Helping Innovators Innovate*. http://www.prod-dev.com/products.shtml. June 2005

[Pro05b] PROJECT MANAGEMENT INSTITUTE. *Project Management Body Of Knowledge Introduction*. http://www.palmettochapter.com/education/presentations/PMBOK%20Guide.ppt. August 2005

[Pro05c] PROJECT MANAGEMENT PARTNERS. *ISO 10006: Risky Business*. http://www.pmpartners.com/resources/iso10006.html. August 2005

[PSI05] PSINET CONSULTING SOLUTIONS. *Quality Concepts within CMM and PMI*. www.exu.ilstu.edu/spin/presentations/HSPIN_Preso020701.ppt. August 2005

[PWOB04] POPP, J. ; WAGENER, A. ; ORTLOFF, D. ; BEUNDER, M.: A Novel Approach Towards Standardization of MEMS Process Development. In: *Proceedings of the 9th International Conference on the Commercialization of Micro and Nano Systems, Edmonton*, 2004. – COMS 2004

[Ree05] REED BUSINESS INFORMATION. *Role of Controls in Product Lifecycle Management*. http://www.manufacturing.net/ctl/article/CA421454. August 2005

[ReV04] REVELLE, Jack B.: *Quality Essentials – A Reference Guide from A to Z*. 1st edition. ASQ Quality Press, 2004. – ISBN 0–87389–618–1

[San05] SANDEEP KAMATH. *Requirements engineering: a roadmap*. http://sunset.usc.edu/classes/cs589_2004/Week6a.ppt. August 2005

[Sea05] SEAPINE SOFTWARE INC. *Seapine TestTrack Pro*. http://www.seapine.com/ttpro.html. August 2005

[Sem05] SEMICONDUCTOR EQUIPMENT AND MATERIALS INTERNATIONAL. *Download Individual SEMI Standards*. http://wps2a.semi.org/wps/portal/_pagr/116/_pa.116/121?docName=P001204. September 2005

[SGCD02] DA SILVA, M. G. ; GIASOLLI, R. ; CUNNINGHAM, S. ; DEROO, D.: MEMS Design for Manufacturability (DFM). In: *Proceedings of IEEE Sensors Expo and Conference, Orlando*, 2002. – Sensors 2002

[Sie05a] SIEBEL SYSTEMS. *Siebel Systems CRM*. http://www.crmondemand.co.uk/. June 2005

[Sie05b] SIEGLAUB, JAY. *PMBOK and PRINCE2*. www.pminyc.org/files/Siegelaub_Oct-00.ppt. August 2005

[Sil05a] SILICON INTEGRATION INITIATIVE, INC. *Silicon Integration Initiative*. http://www.si2.org/. September 2005

[SIL05b] SILVACO INTERNATIONAL. *SILVACO - EDA Software Connecting TCAD to Tapeout*. http://www.silvaco.com/. November 2005

[Sof02] SOFTWARE ENGINEERING INSTITUTE. *Bibliography on Integrated Product and Process Development CMMI-SE/SW/IPPD*. http://www.sei.cmu.edu/cmmi/publications/ippd-biblio.pdf. Janurary 2002

[Sof05a] SOFTWARE ENGINEERING INSTITUTE. *Capability Maturity Model*. http://aelinik.free.fr/cmm/default.htm. September 2005

[Sof05b] SOFTWARE ENGINEERING INSTITUTE. *Level 3 - Software Product Engineering*. http://aelinik.free.fr/cmm/tr25_l3e.html. June 2005

[SOH+06] SCHMIDT, T. ; ORTLOFF, D. ; HAHN, K. ; WAGENER, A. ; POPP, J. ; BRÜCK, R.: Management and Verification for MEMS Fabrication Processes. In: *Proceedings of SPIE: Micromachining and Microfabrication Process Technology X, San Jose* Bd. 6109, 2006. – Photonics Europe 2006

[Sou05] SOURCEFORGE. *TightVNC*. http://www.tightvnc.com/. July 2005

[Sou06] SOUTHWEST CENTER FOR MICROSYSTEMS EDUCATION. *SCME Industry Survey*. http://www.scme-nm.org/. January 2006

[Spa05] SPARX SYSTEMS PTY LTD. *Enterprise Architect*. http://sparxsystems.com.au/products/ea.html. August 2005

[SPI05] SPIRIT CONSORTIUM. *Structure for Packaging, Integrating and Re-using IP within Tool-flows*. http://www.spiritconsortium.org/. December 2005

[Spo05] SPOCE LTD. *PRINCE2 Introduction*. http://www.spoce.com/PRINCE2%20P2Introduction.htm. August 2005

[Sta05a] STAGE-GATE, INC. *Giving wings to product innovation*. http://www.stage-gate.com/index.html. June 2005

[Sta05b] STATSOFT. *Experimental Design*. http://www.statsoft.com/textbook/stexdes.html. November 2005

[Sta05c] STATSOFT INC. *Statistica Suite*. http://www.statsoftinc.com/products/products.htm. August 2005

[Sub05] SUBVERSION HOMEPAGE. *Subversion*. http://subversion.tigris.org/. December 2005

[Sun04a] SUN MICROSYSTEMS INC. *Java 2 Platform, Enterprise Edition (J2EE) Homepage*. http://java.sun.com/j2ee/. November 2004

[Sun04b] SUN MICROSYSTEMS INC. *Java Web Start Homepage*. http://java.sun.com/products/webstart/. November 2004

[SUN05a] SUN. *Code Conventions for the Java Programming Language*. http://java.sun.com/docs/codeconv/. August 2005

[SUN05b] SUN. *Java Look and Feel Design Guidelines, Second Edition*. http://java.sun.com/products/jlf/ed2/guidelines.html. August 2005

[Sys05a] SYSTAT SOFTWARE INC. *SYSTAT*. http://www.systat.com/products/systat/. August 2005

[Sys05b] SYSTEMS, Cadence D. *Cadence Design Systems Website*. www.cadence.com. August 2005

[Tav06] TAVARES, Silvia. *Project Management Processes and Quality Checklist*. http://www.auditnet.org//docs/PM-AuditQuestionnaire.doc. January 2006

[Tel05] TELELOGIC AB. *Telelogic DOORS - Requirements Management*. http://www2.telelogic.com/products/doorsers/index.cfm. August 2005

[Tex05] TEXAS TECH UNIVERSITY. *PSPE - Program for Semiconductor Product Engineering*. http://www.ee.ttu.edu/pspe/. June 2005

[THE95] THE WELCH COMPANY. *Reviewed ISO 10006 on Project Management, compared PMBOK*. http://www.welchco.com/sd/08/00101/02/95/07/21/134937.HTM. August 1995

[Tho05] THOMAS, STEVEN. *An Agile Comparison*. http://www.balagan.org.uk/work/agile_comparison.htm. March 2005

[UGS05] UGS. *Teamcenter Engineering*. http://www.ugs.com/products/teamcenter/sol_prod/engineering/. June 2005

[Ulr03] ULRICH, Karl T.: *Product Design and Development*. 1st edition. McGraw Hill, 2003. – ISBN 007–123273–7

[VCX05] VCX SOFTWARE LTD. *The IP Portal*. http://www.thevcx.com/home.html. December 2005

[Ver05] VERPOORTEN, JAN HERMAN. *Projectmanagement – PRINCE2 versus PMBOK*. http://www.cs.uu.nl/docs/vakken/pm/sheets/college7.pdf. August 2005

[VTT01] VTT RESEARCH LABS. *Virtual prototyping – VTT Research Programme*. http://www.vtt.fi/vtt/tutkimus/virtuaaliprototypointi/loppuraportti/s210tekstirp.pdf. February 2001

[Wag05] WAGENER, Andreas: *Fertigungsnahe Entwurfsunterstützung für die Mikrosystemtechnik*, Universität Siegen, PhD thesis, 2005

[Web05a] WEBEX. *WebEx Web conferencing*. http://www.webex.com/. July 2005

[WEB05b] WEBSINA, INC. *Bugzero - Bug/Defect/Issue Tracking System*. http://www.websina.com/bugzero/index.html. August 2005

[Wid05] WIDEMAN, MAX. *Comparing PRINCE2 with PMBOK*. http://www.pmforum.org/library/papers/2002/Prince2vsGuide3.pdf. August 2005

[Wik05a] WIKIPEDIA. *Capability Maturity Model*. http://en.wikipedia.org/wiki/CMMI. August 2005

[Wik05b] WIKIPEDIA. *Change Management*. http://en.wikipedia.org/wiki/Change_Management. June 2005

[Wik05c] WIKIPEDIA. *Change Management (ITIL)*. http://en.wikipedia.org/wiki/Change_Management_(ITIL). June 2005

[Wik05d] WIKIPEDIA. *Configuration Management*. http://en.wikipedia.org/wiki/Configuration_management. June 2005

[Wik05e] WIKIPEDIA. *Issue tracking system*. http://en.wikipedia.org/wiki/Issue_tracking_system. June 2005

[Wik05f] WIKIPEDIA. *IT Infrastruture Library (ITIL)*. http://en.wikipedia.org/wiki/ITIL. June 2005

[Wik05g] WIKIPEDIA. *New product development*. http://en.wikipedia.org/wiki/New_product_development. September 2005

[Wik05h] WIKIPEDIA. *PRINCE2 - or PRojects IN a Controlled Environment*. http://en.wikipedia.org/wiki/PRINCE2. June 2005

[Wik05i] WIKIPEDIA. *Product (business)*. http://en.wikipedia.org/wiki/Product_%28business%29. November 2005

[Wik05j] WIKIPEDIA. *Project management*. http://en.wikipedia.org/wiki/Project_management. August 2005

[Wik05k] WIKIPEDIA. *Requirements analysis*. http://en.wikipedia.org/wiki/Requirements_engineering. August 2005

[Wik05l] WIKIPEDIA. *Unix-Shells*. http://de.wikipedia.org/wiki/Unix-Shell. July 2005

[Wor04] WORLD WIDE WEB CONSORTIUM (W3C). *Extensible Markup Language (XML) Homepage*. http://www.w3.org/XML/. November 2004

[WPH03] WAGENER, A. ; POPP, J. ; HAHN, K.: PRINCE - Process Information and Management Center. In: *Proceedings Micro System Technologies 2003, München*, 2003. – MST 2003

[WPS+05] WAGENER, A. ; POPP, J. ; SCHMIDT, T. ; HAHN, K. ; BRÜCK, R. ; ORTLOFF, D.: Environment for Design and Verification of MEMS Fabrication Processes. In: *Proceedings of the MST 2005*, 2005. – MST 2005

[WSP+06] WAGENER, A. ; SCHMIDT, T. ; POPP, J. ; HAHN, K. ; BRÜCK, R. ; ORTLOFF, D.: Process Design and Tracking Support for MEMS. In: *Proceedings of SPIE: Micromachining and Microfabrication Process Technology X, San Jose* Bd. 6109, 2006. – Photonics West 2006

[Zav97] ZAVE, Pamela: Classification of research efforts in requirements engineering. In: *ACM Comput. Surv.* 29 (1997), Nr. 4, S. 315–321. – ISSN 0360–0300

A. Details for Managing a project

Appendix A gives more details about the project management sequence, the steps of the business processes and their inputs and outputs. The sequences and steps are basically taken from the PRINCE2 standard, partly extended or slightly rearranged. The input and output are taken from PRINCE2 as well, while a full set of templates for all project management and product development artefacts has been created. This set contains all templates advised by the PRINCE2 standard with additions of further predefined deliverables. This set is contained within the default electronic project binder. The set of templates and the electronic project binder is defined in more detail in section A.3

A.1. The Management of the project preparations

The project preparations are done in Stage 2 of the development method (chapter 3.4). In detail the stage performs the steps summarised in table A.1. The table shows the inputs and outputs of the activities. They are graphically presented in figure 3.9 and figure A.1. An item listed as input and output becomes updated during the activity. Besides external inputs or outputs, all inputs and outputs are retrieved and stored from and to the electronic project binder (see below) referenced as ”Management Information” in the drawings. The activities are executed in the sequence of the table rows and in accordance to the PRINCE2 best practices (e.g. [GC02] and [Ben02]). The flow of business process steps can be seen in figure 3.11 as well. The abbreviations SU, IP, DP, PL, CS, MP, SB and CP followed by a number refer to the corresponding PRINCE2 process groups and the step number. They stand for SU = Starting Up a project, IP = Initiating a Project, DP = Directing a Project, PL = Planning, CS = Controlling a Stage, MP = Managing Product delivery, SB = Managing Stage Boundaries and CP = Closing a Project.

Inputs	Activities	Outputs
Project Mandate	SU1 Appointing a Project Board Executive and Project Manager Set up the electronic project binder	Agreed Executive job description Agreed Project Manager job descr. Electronic project binder
Agr. Exec. and PMngr. job descr. Project Mandate	SU2 Designing a Project Management Team	Draft job description PM team structure
Draft job descritpion PM team structure	SU3 Appointing a Project Management Team	Agreed job descriptions

Inputs	Activities	Outputs
Project Mandate	SU4 Preparing a Project Brief	Project Brief Risk Log
Project Brief Risk Log	SU5 Defining Project Approach	Project Approach
Project Approach Project Brief Risk Log	SU6 Planning an Initiation Stage(using the process sequence in table A.2 as subprocess)	Draft Initiation Plan Risk Log
Draft Initiation Plan Risk Log Job descriptions PM team structure Project approach Project Brief	DP1 Authorising Initiation	Initiation Stage Plan Approved Project Brief Project Brief
Authorisation to proceed Project Brief Project Approach Quality Standards from Supplier and/or customer QMS	IP1 Planning Quality	Project Quality Plan Quality Log
Project Brief Project Approach Project Quality Plan Risk Log	IP2 Planning a Project(using the process sequence in table A.2 as subprocess)	Project Plan Risk Log
Project Brief Project Approach Risk Log Project Plan	IP3 Refining the Business Case and Risks	Business Case Risk Log Project Plan
Project Quality Plan Risk Log Project Plan	IP4 Setting up Project Controls	Communication Plan Project controls Risk Log Project Plan
Project Plan Project Quality Plan	IP5 Setting up Project Files	Issue Log Lessons Learned Log Project Quality Plan
Project Brief PM team structure and job descr. Project Approach Project Quality Plan Project Plan Business Case Risk Log Project controls Communication Plan	IP6 Assembling a Project Initiation Document including planning the next stage(using the process sequence in table A.4 as subprocess)	Draft PID Next Stage Plan

Table A.1.: Stage 2 activities with inputs and outputs

A.2. The Management of the R&D loop

Like the management of the other method aspects, the Management of the R&D loop is performed according to the PRINCE2 project management standard. Because of that the following descriptions are rather brief to not replicate PRINCE2 literature like [GC02] or [Ben02]. To guide the project managers and product engineering efforts the overall flow and sequence of process steps is

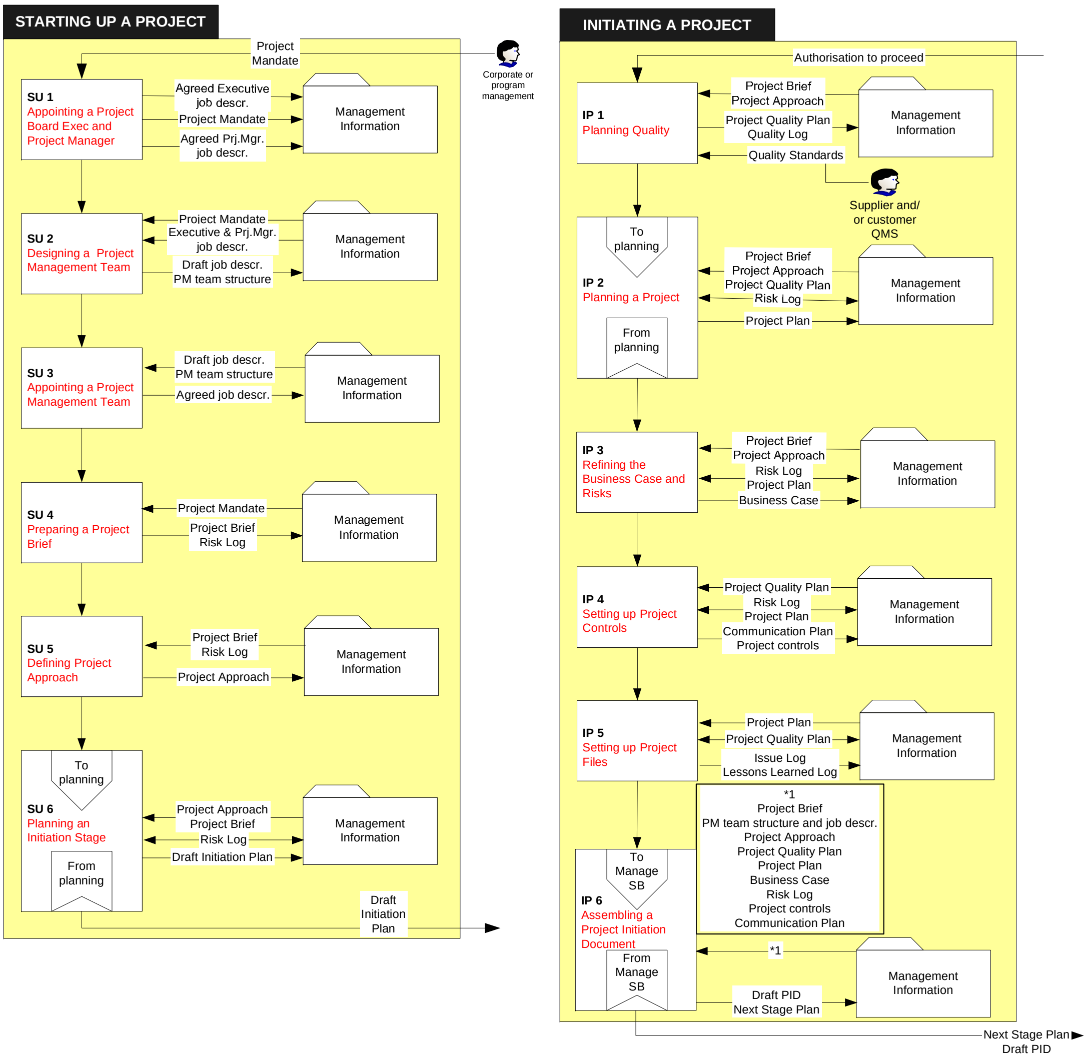


Figure A.1.: Prince2 process Starting up a project and Initiating a project

presented. More information about the details of the different process groups, process steps and activities can be extracted from literature.

The management process steps can be grouped into certain process groups like planning activities, managing stage boundaries etc. This grouping is already included in the PRINCE2 standard and can be seen in figure 3.9. Apart from this grouping a more flow oriented grouping (preparing a project, planning in various forms and steps, inside a stage loop, finalizing a stage, exception treatment, gate assessment and finishing a project) can be set up. This structure is used when describing and depicting the serialized method. It can be seen in figure 3.10. The following sections will describe these flow oriented process groups in a bit more detail.

Process group Planning Some sequences of process steps can be considered as modules which are plugged into or used by process steps of other groups. One of these generic activity sequences are the planning steps which are used to generate the overall project plan, derive stage plans, plan a work package etc. A general introduction into the planning efforts and the importance of the planning and tracking is given in section 4.10. The steps of a planning subprocess are presented in table A.2. The table shows the different steps to create a plan showing the required inputs and outputs. A graphical representation of the process sequence is given in figure A.2. More details about the planning activities and its best practices and how a good plan can be created are given in section 4.10 and in e.g. [GC02] and [Ben02].

For the planning it is important that the deliverables based planning paradigm of PRINCE2 is applied. In summary this means that not the activities are the start for the planning efforts, the deliverables (products in PRINCE2 terms) for the next gate are the starting point. These deliverables and their development are the highest level in the project plan and are successively broken down into work packages, activities for the work packages and tasks. To achieve this breakdown initially a Product Breakdown Structure (PBS) is created. This PBS recursively splits down the end product into smaller deliverables (divide-and-conquer strategy). Each deliverable (on all levels) needs to be documented using the Deliverable Description template (PRINCE2) building the contract between the different involved parties. These descriptions build the basis for creating the Work Package Descriptions (template) used to execute the development efforts. From the deliverables list the Product Flow Diagram (PFD), a dependency graph, can be created. This graph enables a proper scheduling and resource assignment via a project planning tool like MS Project. More details about the product/deliverables based planning approach can be found in [GC02] and [Ben02].

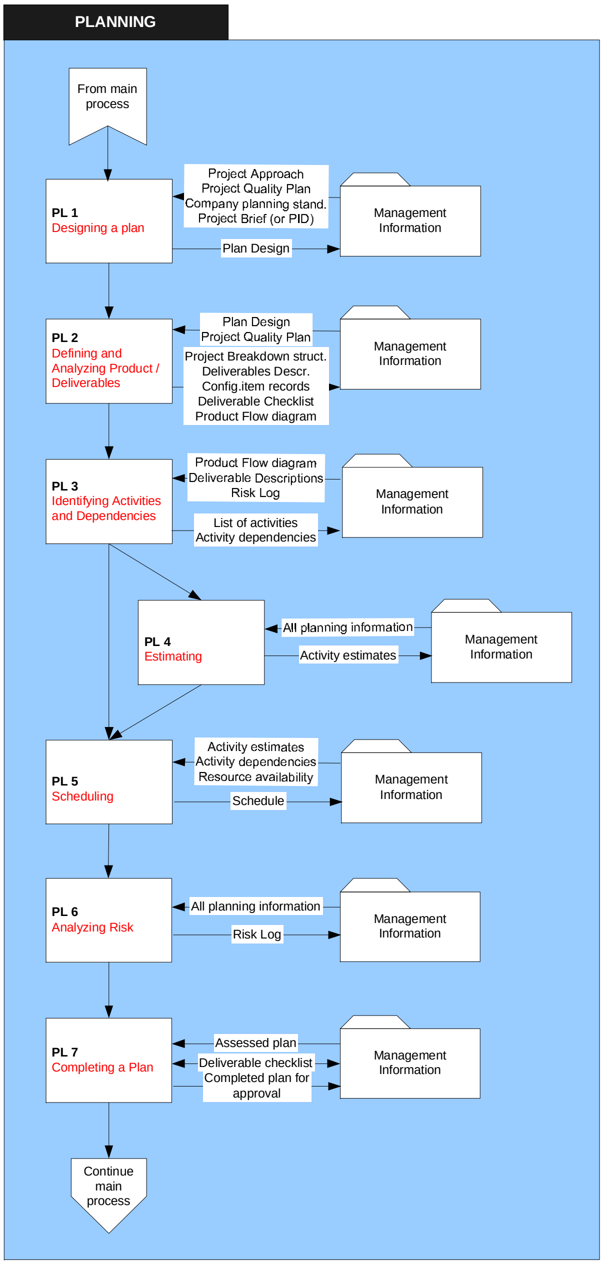


Figure A.2.: Planning process sequence

Inputs	Activities	Outputs
Project Approach Project Quality Plan Company planning standards Project Brief (or PID)	PL1 Designing a plan	Plan Design
Plan Design Project Quality Plan	PL2 Defining and Analyzing Product/Deliverables	Project Breakdown Structure(PBS) Deliverable Descriptions Configuration item records Product/Deliverables Checklist Product Flow diagram
Product Flow diagram Deliverables Descriptions Risk Log	PL3 Identifying Activities and Dependencies	List of activities Activity dependencies
All planning information	PL4 Estimating (optional and only used where applicable)	Activity estimates
Activity estimates Activity dependencies Resource availability	PL5 Scheduling	Schedule
All planning information	PL6 Analyzing Risk	Updated Risk Log
Assessed plan Product/Deliverable checklist	PL7 Completing a Plan	Product/Deliverable checklist Completed plan for approval

Table A.2.: Planning activities with inputs and outputs; used in several places

Process group Inside a stage loop The main development activities for a project (technical, marketing&sales, financial, legal, etc.) are managed by the R&D stage loop highlighted with the yellow box in figure 3.15. Here only the managerial steps and artefacts are highlighted, the technical, financial, etc. deliverables are described within the individual stages and gates.

The process steps of this group authorize the different work packages and manage the execution of the work package (including planning the work package). The process steps have to handle internal and external issues or exceptions and keep track of the project status and progress and report about it. Necessary corrective actions are initiated and their progress is coordinated with the projects executive. More details about the input and output of the different processes are presented in table A.3. The table also gives more details about the conditional executions of this loop. Figure A.3 presents the parts of the Prince2 standard responsible for driving the work.

Inputs	Activities	Outputs
Authorisation to proceed Approved Stage or Exception Plan Deliverable Descriptions Configuration Item Record Risk Log	CS1 Authorising Work Package	Approved Stage or Exception Plan Configuration Item Record Risk Log
Team Plan Risk Log	MP1 Accepting a Work Package	Updated Team Plan Updated Risk Log Authorised Work Package
Authorised Work Package	MP2 Executing Work Package	Completed Work Package Checkpoint report Updated Team Plan Updated Risk Log

Inputs	Activities	Outputs
Completed Work Package	MP3 Delivering a Work Package	Approved Work Package
Approved Work Package Configuration Item Records	CS9 Receiving Completed Work Package	Configuration Item Records Work Package status
Work Package status Checkpoint Report Quality Log Stage Plan	CS2 Assessing Progress	Stage Plan Checkpoint Report (to Management Information and Project Archive) Stage status information
Issue Log New Project Issues	CS3 Capturing Project Issues (optional only if triggered by new issues from log or customer)	Updated Issue Log
Business Case Project Plan Stage Plan Risk Log	CS4 Examining Project Issues(optional only if triggered by new issues from log or customer)	Updated Risk Log Updated Issue Log
Stage status information Issue Log Risk Log Stage Plan Project Plan Business Case Configuration Item Record	CS5 Reviewing Stage Status	(no direct output only decision where to continue as below)
Stage status information Issue Log Risk Log Stage Plan Quality Log Communication Plan	CS6 Reporting Highlights	Highlight Report Stage status information(to archive)
depending on the Stage status either of:		
	Stage not finished: trigger CS1 in this table to continue work	
	Stage finished: trigger to Finalizing a stage in table A.4	
	plan deviation detected: continue with CS7 exception treatment table A.6	
	tolerance threat detected: continue with CS8 in table A.6	
	all projects work done: trigger to Close Project in table A.7	

Table A.3.: Inside a R&D loop activities with inputs and outputs

Process group Finalizing a stage After all work packages and deliverables of a stage are successfully completed, the process steps of finalizing a stage are executed. The steps of this subprocess basically round up the current stage and prepare the next stage. For that a plan for the next stage is produced and the overall project plan, the business case and the issue+risk logs become updated. Finally a stage end report triggering the next gate assessment is gen-

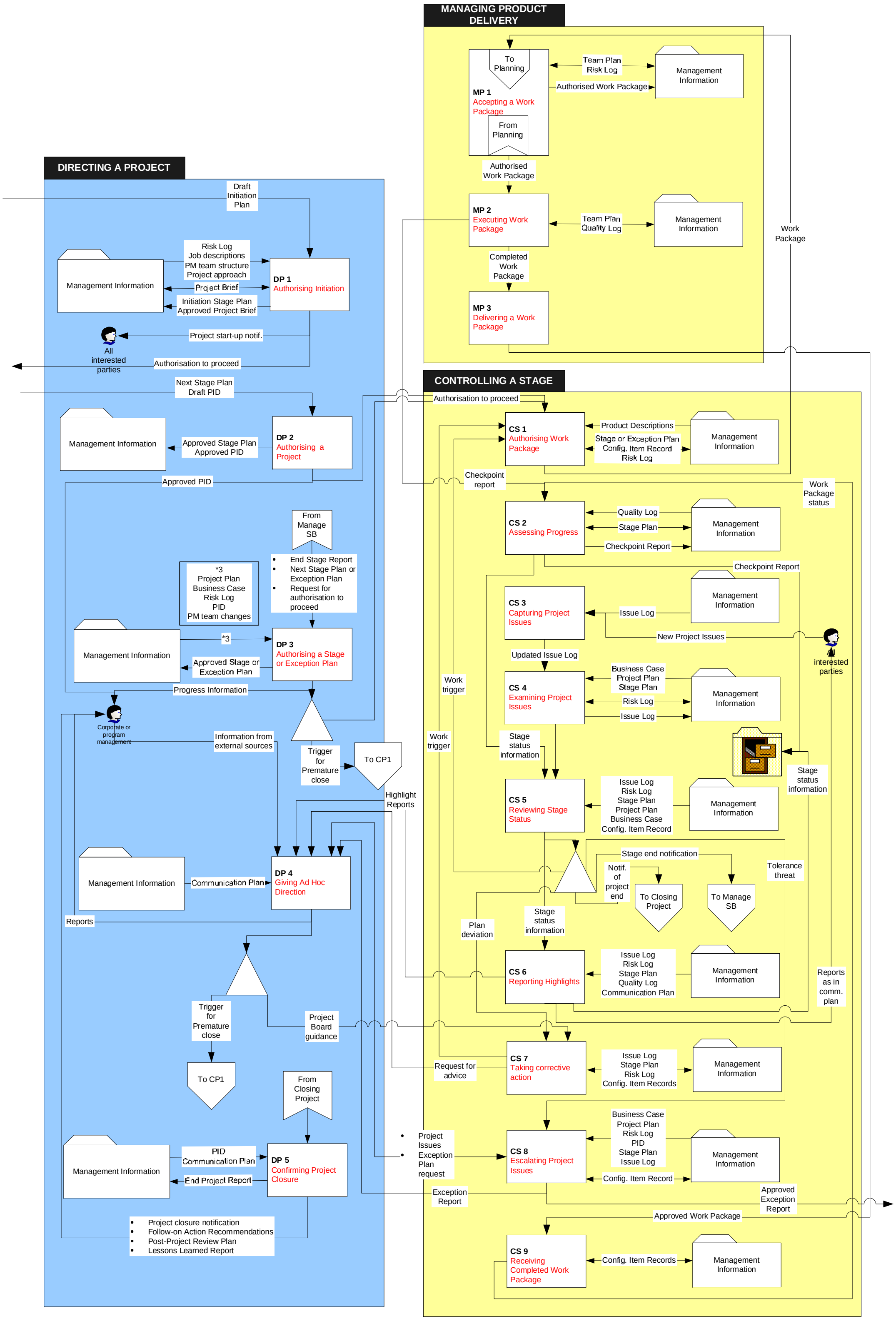


Figure A.3.: Prince2 process Directing a project, managing product delivery and Controlling a stage

erated. More details about the input and output of the different processes are presented in table A.4.

Inputs	Activities	Outputs
Stage end notification Current Stage Plan Project Initiation Document(PID) Issue Log Risk Log PM team structure	SB1 Planning a Stage (using the process sequence in table A.2)	PM team structure Next Stage Plan
Current Stage Plan Next Stage or Exception Plan Project Approach Issue Log Project Quality Plan Project Plan	SB2 Updating a Project Plan (using the process sequence in table A.2)	Next Stage or Exception Plan Project Approach Issue Log Project Quality Plan Project Plan
Project Plan Issue Log Next Stage or Exception Plan Business Case Risk Log	SB3 Updating a Project Business Case	Next Stage or Exception Plan Business Case Risk Log
Business Case Next Stage or Exception Plan Project Plan Issue + Risk Log	SB4 Updating the Risk Log	Next Stage or Exception Plan Project Plan Issue + Risk Log
Current Stage Plan Business Case Issue Log Risk Log Quality Log Communication Plan Next Stage or Exception Plan Config. Item Records Lessons Learned Log	SB5 Reporting Stage End	Next Stage or Exception Plan Config. Item Records Lessons Learned Log Request for authorisation to proceed End Stage Report
depending on context either:		
	continue IP6	
	trigger DP3 in table A.5	

Table A.4.: Finalizing a Stage activities with inputs and outputs; used in several places

Process group Gate assessment This process group, actually only one process step and decision making, is the key for successful product developments. As outlined in the description of the product development method, e.g. in section 3.4.1, the gates are the essential check points during the lifetime of a project. They are essential to distinguish between the winning and failing project ideas. The importance and procedure of the gate assessments is described in 3.4.1 and should not be re-discussed here. Only the managerial inputs and outputs are listed in table A.5.

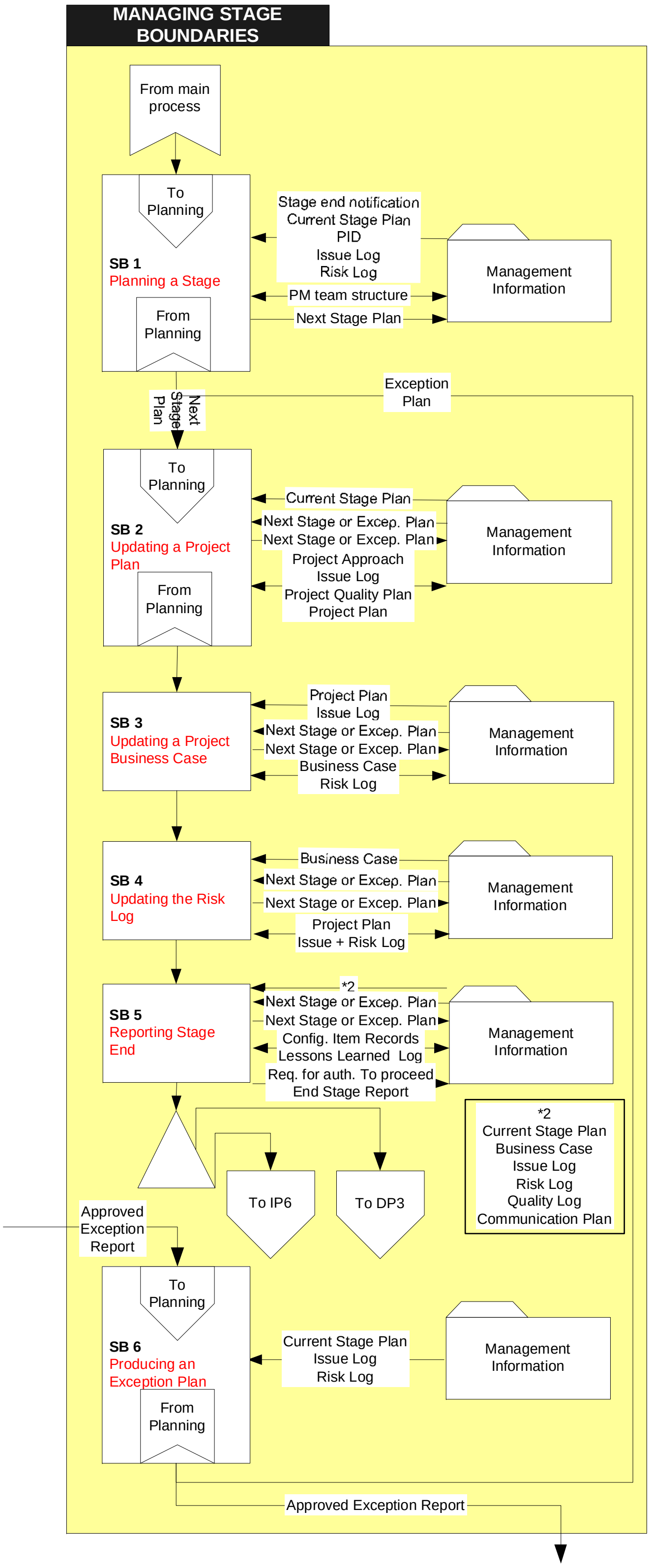


Figure A.4.: Prince2 process Managing Stage Boundaries

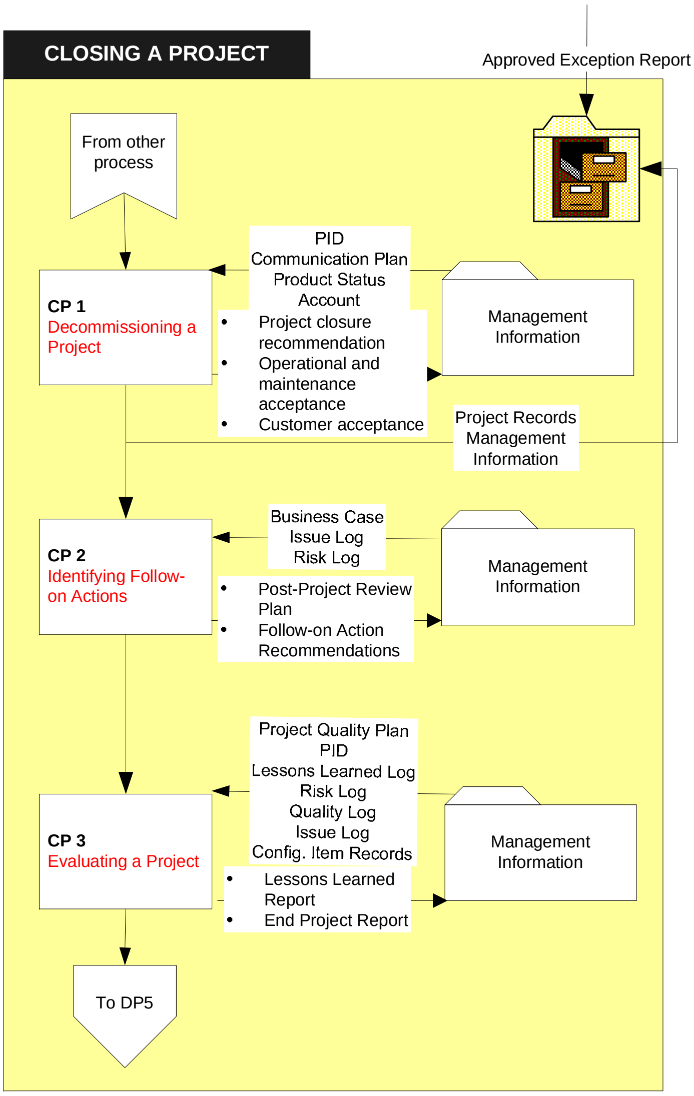


Figure A.5.: Prince2 process Closing a Project

Inputs	Activities	Outputs
End Stage Report Next Stage Plan or Exception Plan Request for authorisation to proceed Project Plan Business Case Risk Log PID PM team changes	DP3 Authorising a Stage or Exception Plan	Approved Stage or Exception Plan Progress Information(to corporate or program management)
depending on decision either:		
	authorization to proceed: trigger CS1 in table A.3	
	Trigger to premature end of project continuing with Close Project in table A.7	

Table A.5.: Performing Gate assessment with inputs and outputs

Process group Exception treatment In the process group Exception treatment the project steering committee becomes actively involved in the project. This committee consists out of the project executive, the project manager and the corporate program management. Usually the project is managed by only the project manager and projects executive. In the exception case, the steering committee gets the role to give project guidance, settle project issues and approve exception plans. On the other hand the steering committee might feed information from external sources into the project, potentially changing the course of the project. In extreme cases the project might be even triggered for a premature end. More details of the process steps to treat exceptions are presented in table A.6 below, together with the steps inputs and outputs. The steps are marked with the red boxes in figure 3.15 and belong to different process groups in the PRINCE2 standard (see step names). In the description here they are grouped together into this new group to mark them as being not necessarily part of a standard project life cycle.

Inputs	Activities	Outputs
Issue Log Stage Plan Risk Log Configuration Item Records	CS7 Taking corrective action	Issue Log Stage Plan Risk Log Configuration Item Records
depending on whether necessary or already given:		
	advice required: trigger DP4 in table A.6	
	no advice required or already given: trigger CS1 in table A.3 to continue work	

Inputs	Activities	Outputs
Communication Plan Information from external sources Highlight Report Request for advice Exception Report Project Issues Exception Plan request	DP4 Giving Ad Hoc Direction	Project Issues Exception Plan request Reports(to corporate or program management)
either:	Trigger to premature end of project continuing with Close Project in table A.7	
	Project Board guidance continuing with CS7 in this table	
Tolerance threat Project Issues Exception Plan request Business Case Project Plan Risk Log PID Stage Plan Issue Log	CS8 Escalating Project Issues	Configuration Item Record Exception Report Approved Exception Report
depending on whether already escalated:		
	unsolvable issues or exception report: trigger DP4 in table A.6	
	exception report approved or exception plan requested: trigger SB6 in table A.6	
Approved Exception Report Current Stage Plan Issue Log Risk Log	SB6 Producing an Exception Plan(using the process sequence in table A.2)	Exception Plan
	continue with SB2 in table A.4	

Table A.6.: Exception treatment activities with inputs and outputs

Process group Finishing the project The process steps of this process group are responsible to cleanly close a project and release the resources, including the personal working on the project. For that it uses the PRINCE2 processes consisting of decommissioning the project team, identifying and documenting the project follow up actions and to evaluate the project. The findings are summarised in a lessons learned report and an end project report. More details about the input and output of the different processes are presented in table A.7. A graphical representation of the PRINCE2 standard of these steps is presented in figure A.5.

Inputs	Activities	Outputs
PID Communication Plan Product Status Account	CP1 Decommissioning a Project	Project closure recommendation Operational and maintenance acceptance Customer acceptance Project Records(to archive) Management Information(to archive)
Business Case Issue Log Risk Log	CP2 Identifying Follow-on Actions	Post-Project Review Plan Follow-on Action Recommendations
Project Quality Plan PID Lessons Learned Log Risk Log Quality Log Issue Log Configuration Item Records	CP3 Evaluating a Project	Lessons Learned Report End Project Report

Table A.7.: Closing a project activities with inputs and outputs

A.3. Project + Product Management templates

This part of the appendix summarises the project and product management templates used by the product development method. In principle this set contains all management templates advised by the PRINCE2 standard with additions of further predefined deliverables. They are arranged in an electronic project binder, presented as ”Management Information” in the drawings. For the creation of these project binders a set of project create scripts have been implemented as documented in 4.5. The binder and project repository created contains a full set of project management templates tailored to the needs of the product development method. The contents of the templates proposed in [GC02] and [Ben02] is adapted and extended to the new scope. Additional templates have been added to cater for not covered areas. The set contains templates for all aspects of project management documentations with additions for the product documentations.

To limit the amount of descriptions and replications of PRINCE2 standard templates, the following sections only describe changed or added templates. The contents of the other templates cover minimally what is described in [GC02] and [Ben02]. All templates received additional standard sections for internal document management and quality assurance aspects. Added comments and hints ease the filling of the templates.

A.3.1. Project Mandate

The Project Mandate template is reproduced from the PRINCE2 standard and covers the aspects of Authority, Background, Project objectives, Scope, Constraints, Interfaces and Quality expectations. Additionally it outlines the Business Case (with reasons), the Project tolerances, the Reference to any associated documents or products. Furthermore it gives an indication of the project Executive and Project Manager and Customer(s), user(s) and any other known interested parties. Apart from that, the standard sections for internal document management and quality assurance aspects are contained in the template.

A.3.2. End Stage report

The End Stage report has been modified with additional explanations and a recommendation for the further course of the project. In there the project team has to propose whether to go, recycle, hold or kill the project. This recommendation summarises the projects internal knowledge and should be the result of a critical self-assessment as input for the next gate.

A.3.3. Requirement Specification (RS)

The one component laying the baseline of a product and its different versions is a requirement specification. To make the requirements collection process thorough and complete, a predefined template has been developed. It guides and supports the requirements elicitation and defines the different angles and perspectives required to evaluate on. The template covers the aspects

- Preface
  Providing a version history with change comments, document approval status and an abbreviations table.
- Introduction
  To provide the product definition.
  - Summary
    Identification of the product by its name and a brief overview about it.
  - Purpose/Objective
    Desired outcome from the Project and/or Product Development goals.
- Terms
  Definition of all terms with unusual, new or special meaning.
- Features
  This is to address the ”what”.
  - Required
    List of required features.
  - Desired
    List of optional features.

  - Performance Goals
    Provides specific performance goals for the product
  - Non-functional requirements
    Provides a list of remaining requirements which are not requesting specific functionality or performance. This could be certain delivery mechanism, look-and-feel, time frames, development approaches e.g. rapid prototyping etc.

- Justification & Benefits
  This is to address the ”Why”.
- Target Customers
  This is to address the ”who” (e.g. Product Applications(s), Target Market, customer Names/Identification).
- Deliverables (with target dates)
  List the specific deliverables. These represent major quantifiable milestones with tangible results and a target for ”When” the milestone must be complete. This includes an MS Project plan on an abstract/milestone level which can be used as a basis for the more detailed planning.
- Competitors & Alternatives
  This provides information on alternative implementations that should be or have been considered and competitor's products that serve as a point of comparison for effective product development.
- Comments & Assumptions
  This section documents any assumptions made and any additional comments. To specify any risks, constraints and issues. Also, specifying any external requirements/dependencies.

This baseline laying specification is stored in the specifications/in of the standard electronic binder (4.5) and is under revision control (4.2) to enable reproducibility and communication.

A.3.4. Product Brief

The vision statement or product brief is created during setting up a development project as part of the development project startup activities. It defines the product scope in terms of the development aim. Its main purpose is to define the vision for marketing and market assessment purposes. For a market driven product the vision statement clarifies which requirements of the requirement specification will be addressed by the product and which are out of the scope of the development project. It clearly defines what should be achieved by the development project and gives an overview about the operation domain.

A.3.5. Product definition

The product definition is kind of an answer of the development organization to the Product Brief and Requirement Specification. It is developed after the initial concept assessments and prototyping tests are performed and defines in detail the characteristics, limitations and implications of the product. The limitations and implications need to clearly address areas of not addressed requirements and implications from the internal perspective (e.g. required development resource) and the customer perspective (e.g. equipment required by customers, customer investments etc.). The product definition or specification defines the development target on a detailed level from several perspectives. For MEMS design and process IP products a minimum set of perspectives needs to cover

- Design data sheet
  explaining most important device/design characteristics of the electrical and structural domain and their interactions.

- Process blue book
  Description of the principle process flow, possibilities, alternatives

- Integration blue book
  Description of how device/process can be integrated/rolled-out

A.3.6. Change Request (CR)

To modify an already existing or defined project or product it is essential to provide a predefined Change Request template. This template covers the aspects

- Introduction
  Short introduction into the change wish

- Terms
  Definition of terms used in the document

- Object to change
  Specification of exactly which object should be changed. Only one object can be changed per CR. If several items of a project need to be changed, separate CRs need to be produced. An exception can be made, if a single change has implications to several items. Then all items, the change has implications to, need to be listed in the References and Implications chapter.

- General reason(s) for change(s)
  Some text describing the reasons of the change

- Detailed changes
  Detailed descriptions of all items which require change including detailed reasoning and proposals how to change it (approaches, etc). These can be ”New requirements”, ”Extensions”, ”Adaptations” etc.

- Prioritisation
  If the change request contains more than one severe change, a prioritisation should be given to ease the determination of change planning.

- Implications
  Here a detailed description of the implications of the change/changes have to be given. This includes a discussion and evaluation of the pros and cons implementing the change versus not implementing the change.

- Discussion
  Here a detailed discussion of the pros and cons of the requested changes have to be given.

For reproducibility and communication reasons the template is included in the electronic project binder in the specifications/in folder.

B. Details of Engineering Framework Blocks and Components

B.1. Configuration Management

According to [Ber01] Configuration Management for software development needs to cover the aspects of Configuration Identification, Configuration Control, Status Accounting Audit and Review. Additionally it needs to contain Build Management, Process Management and Team Work. Because these points are so important, they are listed below again in their original form.

- *Configuration Identification,*
  which includes determining which body of source code you are working with. This makes it possible to know, among other things, that you are fixing a bug in the source code which is in the correct release.

- *Configuration Control,*
  controlling the release of a product and changes to it throughout the life cycle to ensure consistent(re) creation of a baseline software product. This can include not only changes to source files, but also which compiler and other tools were used so issues such as differences between compiler support for language features can be taken into account.

- *Status Accounting Audit,*
  recording and reporting the status of components and change requests, and gathering vital statistics about components in the product. One question we may want to answer is:, "How many files were affected by fixing this one bug?"

- *Review.*
  validating the completeness of a product and maintaining consistency among the components by ensuring that components are in an appropriate state throughout the entire project life cycle and that the product is a well-defined collection of components.

It can also be said to include

- *Build Management,*
  ie, managing what processes and tools developers use to create a release, so it can be repeated.

- *Process Management,*
  ensuring that the organizations development processes are followed by those developing and releasing the software

- *Team Work,*
  controlling the interactions of all the developers working together on a product so that peoples changes get inserted into the system in a timely fashion.

B.2. Design setup details

Because EDA tool environments typically work on Unix/Solaris/Linux based systems, script based environment settings are used typically. For choosing a scripting language several possibilities exist. A good overview about Unix-Shells is given in [Wik05l]. A widespread used option is the GNU bash [Fre05a] because it is more powerful and more modern than other shells. For setting up a common design tool environment it is essential that all engineers working on the same project have identical settings. They need to work in their own work space build out of a repository sandbox. To achieve the identical setup a centrally managed project initialization environment needs to be set up. Therefore a project initialization environment out of several scripts and tools has been created. The central entry point to this environment is one script (called set_project) which performs all required settings for that project. The settings can be defined by the projects tool smith by defining a project initialization file using the other tools of the project environment. The initialization file is executed by the set_project script to perform all required settings. The environment provides functions for:

- Setting the environment for a given tool
  This function takes an initialization file as parameter and performs all required settings for that specific tool and version. Different versions of the same tool might require a separate initialization file because the different tools version need to be installed in parallel.

- Setting the technology to be used by the project.

- Setting environment variables with additional checks.

- Setting and cleaning up the search path.

- Presenting status evaluation results on the Intranet.

B.3. Discussion Tracking

The tracking of development discussions and discussion with external sources e.g. customers are a valuable source of knowledge as well. Therefore these discussions should be captured as well. For this capturing several approaches can be used:

- CRM approach
  This approach uses the customer interaction log of a Customer Relationship Management (CRM) System. Precondition is that there is a CRM system already available and that it supports customer interaction logs. Potential drawbacks of this approach are that the systems are normally not directly designed to support questions and answers and that the integration and general access might be limited.

- Issue Tracking System Approach (ITS)
  This approach uses an Issue Tracking System or Trouble Ticket System to track the discussions. The drawback of this approach is that the usage of these systems might impose more integration efforts and that they typically do not support full text searches in attachments.

- Document Management System approach (DMS)
  This approach uses a Document Management system to keep track of discussion knowledge. The major drawback with this approach is the limited support for direct questions and answers and that the integration into the environment is not always provided or straight forward.

- FAQ approach
  The Frequently Asked Question (FAQ) approach initially looks like a straight forward solution and the best fit. After detailed evaluation it can be seen that this approach normally requires major editorial work and that typically the attachment of additional documents is not supported.

The above list only presents a first assessment and gives the major negative points. To systematically assess the possibilities the table B.1 collects and prioritizes the requirements for a discussion tracking system. Matching the requirements against the different approaches delivers the results presented on the right hand side columns of the table:

Priority	**Requirement**	**CRM**	**ITS**	**DMS**	**FAQ**
MH	Central storage space/reference base	+	++	++	++
MH	Accessible by internally everybody/almost everybody	-	++	++	++
MH	Support hierarchy of categories	?	++	++	+
MH	Provide means for questions and answers	O	++	O	++
MH	Document linking possible	++	+	++	+

Priority	Requirement	CRM	ITS	DMS	FAQ
MH	Searching in questions and answers	?	O	O	++
MH	extendable, maintainable and migrateable	-	++	+	++
SH	workflow component for answering and tracking	+	++	+	-
SH	full text searching, even in attachments	?	O	++	O
SH	integrated into infrastructure/Intranet	-	+	+	++
NH	accissible (partly/controlled) to external people	O	+	+	+
MH = Must-Have; SH = Should-Have; NH = Nice-to-Have					
++ = completely fulfilled; + = fulfilled; O = partly fulfilled; - = not fulfilled; ? = questionable					

Table B.1.: Discussion Tracking Requirements vs. approach fulfillment matrix

From table B.1 it can be seen that a combination of the Issue Tracking and FAQ approach would be the best fit. Therefore a system would be beneficial that supports both areas in an easy to use manner. Searching for such a system the Issue Tracking Systems listed in section 4.8.3 can be used for that purpose. Especially the SupportCenter component integrated into a Mambo CMS Intranet solution seems appealing.

As introduced in the discussion about Issue Tracking Systems in section 4.8.3, SupportCenter is a component for the Mambo Open Source Content Management System. The system is intended as a solution for customer support activities including email, telephone and web support. SupportCenter is an advanced helpdesk system, also known as case or issue management system. The systems can be easily customized and has a flexible and powerful workflow system. Its ticketing system supports a dashboard view with filtering and search capability, extensive ticket detail including ability to add replies, notes, time spent and task tracking to each ticket. The automatic and manual ticket assignment and the file attachments support allows to get detailed insights. Next to the ticketing, the system supports knowledge bases. They include categories support, support for usability, rating by users and intensive searching mechanisms. An especially useful feature is the possibility to convert resolved trouble tickets into knowledge base articles. This way resolutions for questions and discussions can be converted into knowledge base articles which can be retrieved back later again. Therefore SupportCenter is useful component for knowledge management.

B.4. Build environment

The central vehicle for all tasks in the Build and update environment is the GNU Make [Fre05c] toolset. The basic idea of the environment is to implement a few different startup scripts performing different tasks like a retrieval from the repository, starting a nightly build and so on. These scripts start one central and generic build make file performing the intended task via its targets. The make file itself delegates the real execution of the task to project dependent makefiles via calling these Makefiles from the predefined locations with predefined targets. To ease the generation of these project dependent makefiles, templates are provided including generic rules and dependencies. These are included by the project dependent makefiles from common files. The delegation mechanism is used to remain as flexible as possible and enable the different projects to perform those tasks during a build process required to determine their status. This is especially useful for the design parts of a project where things like

- Complete Config. Management retrieval, potentially including the tools
- Complete design reassembles by running a Synthesis, Place&Route, DRC, LVS, etc.
- Analysis of the results and
- Presentation on an Intranet page might be required.

Similar requirements are to be fulfilled for software projects where a complete clean, checkout, compile, link and bind might be required to rebuild a software project.