Research School for Operations
Management and Logistics

Product Data Management as enabler for Concurrent Engineering

Controlling the flow of preliminary information in product development

Remko W. Helms

Product Data Management as enabler for Concurrent Engineering

Eindhoven University of Technology, 2002

Cover: R. van Tiggelen, Funcke *Ontwerpers*

Press: Universiteitsdrukkerij, Eindhoven University of Technology

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Helms, Remko Willem

Product data management as enabler for concurrent engineering / by Remko Willem Helms. –
Eindhoven: Technische Universiteit Eindhoven, 2002. -
Proefschrift.-

ISBN 90-386-1787-9

NUGI 684

Keywords: product development / NPD / concurrent engineering / product data management / PDM / engineering data management / EDM / preliminary information / control complexity / preliminary document versions

Ordering information: Electronic copies of this dissertation can be obtained by sending an e-mail to remko@helms.net, electronic copies are free of charge.

Product Data Management as enabler for Concurrent Engineering

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 12 juni 2002 om 16.00 uur

door

Remko Willem Helms

geboren te Amerongen

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir J.C. Wortmann
en
prof.dr.ir. A.C. Brombacher

Copromotor:
dr.ir. H.J. Pels

“ok moi”

Preface

After finishing my masters degree there were two options, work as a consultant for MiQ (formerly M.I.S. Organisatie-ingenieurs) and work as a researcher for the Eindhoven University of Technology. Looking back I am grateful to Henk Jan Pels and Dick Mandemaker that we could work out a proposal in which we could combine the best-of-both-worlds. On one hand there was living in the fast lane as a consultant and at the other there was time for reflection as a researcher.

But right now I am glad that my research came to an end and my thesis is finally there. Firstly, the combination research/consultant required a lot of energy and attention. Therefore I owe many thanks to my girlfriend Pien Zonneveld, for her full support wherever possible. Secondly, for the last one and a half year friends, family, colleagues, and business associates have been asking when I would defend my thesis. Their interest in my research (or is it the party at the moment I receive my degree?) kept me going for which I am thankful.

But finishing my research also means that there comes an end to the regular discussions with my supervisors Hans Wortmann, Aarnout Brombacher, and Henk Jan Pels. Especially those on the document lifecycle and the simulation model with Henk Jan inspired me. There were also interesting discussions with students that I partly supervised and those with Rob van den Elzen in particular.
Of course there are many more people that contributed to my research by means of discussions and reflection. But I would especially like to thank my colleagues at MiQ, my fellow AIO's at the university and the participants in the RapidPDM project. Besides inspiring discussions you also need distraction, therefore I thank my fellow AIO's Jacques Bouman, Frans Mouws, Frank Berkers and my friends and family for doing this.

Finally I would like to thank two people because without them this thesis would never be the way it is right now. First there is Kate Dunne (CIMRU, Ireland) who I met during the RapidPDM project and who offered me to proofread my thesis and corrected my English wherever possible. Second there is Richard van Tiggelen (Ontwerpersgroep Joris Funcke) who made the design for the cover of this thesis.

Contents

1 Introduction

1.1 Background

Today, companies find themselves in a continuous battle for survival. This battlefield is no longer within the region of their location, but encloses the world. Companies have competitors, customers and suppliers around the globe. In order to survive companies have to become better. For a long time the focus was on improving the production process instead of the product development process because it is responsible for the majority of the cost, *i.e.* respectively 80%-20% (Stevens, 1993).

While firms have developed new products since the Industrial Revolution, the importance of doing product development well has increased dramatically in the nineties (Schilling & Hill, 1998). Changes in competition, customer demands, and technology are the forces that have created a competitive imperative for speed, efficiency and high quality in the development process (Wheelwright & Clark, 1993). Moreover, companies are dependent for more than 50 percent of their annual turnover on products that were introduced within in the last five years (Cooper, 1993). These facts contributed to the increasing awareness of the importance of product development. Therefore in many companies the attention has shifted from production in the seventies and eighties to product development in the nineties.

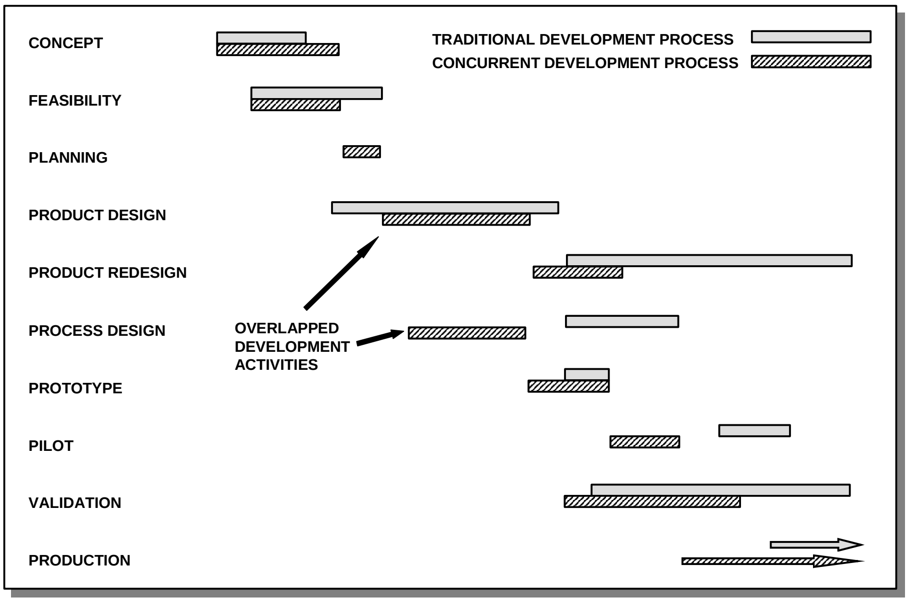


Figure 1 – Overlapped product and process engineering processes

This research focuses on one particular aspect of improving product development processes: cycle time. Companies constantly try to improve their cycle time in order to develop products faster (Eppinger *et al.*, 1997; Reinertsen, 1997; Wheelwright & Clark, 1993). There are several ways to speed up the product development process. First of all it can be achieved by means of better planning the product development process. By studying the sequence in which product development activities are executed it is possible to optimise this sequence. One technique is to conduct activities in parallel as far as possible. Secondly a faster development process can be achieved by optimising the process without fundamentally changing the way products are developed. Techniques that can be used in this context involve simplifying activities, eliminating activities, accelerating activities, and eliminating delays (Millson *et al.*, 1992). The last option, to develop products faster, is to fundamentally change the way products are developed. A widely adopted concept in this context is Concurrent Engineering (Takeuchi & Nonaka, 1986; Blackburn, 1991; Hartley, 1992; Wheelwright & Clark, 1993; Floyd *et al.*, 1993; Ha & Porteus, 1995; De Graaf, 1995; Smith, 1997; Loch & Terwiesch, 1998; Hanssen, 2000).

The main objective of Concurrent Engineering is to shorten the development time of the product by overlapping phases in the product development process. Initially it involved the overlap of product and process engineering processes (see Figure 1). Later it also involved overlapping other phases of the product lifecycle such as operational use and service. The US Department of Defence stimulated this l*ifecycle integration* thinking by means of the CALS (Continuous Lifecycle Acquisition and Support) initiative. A concurrent or integrated product development process is a totally different ball game than the traditional sequential product development process, which is like a relay race where one group of functional specialists passes the baton to the next group. A concurrent development process is more like a rugby game in which a hand-picked multidisciplinary team travels the distance as a unit (Eppinger *et al.*, 1997).

Clark and Fujimoto (1991) concluded that a concurrent development process requires frequent, face-to-face, bilateral communication of preliminary information instead of late release of complete information (which is the case in a sequential process). Companies apply multidisciplinary teams to facilitate close co-operation and the exchange of preliminary information. These multidisciplinary teams consist of representatives from different phases of the product lifecycle. There has been a lot of research with respect to these disciplinary teams. But there has been considerably less research on how to manage incomplete or preliminary information within these multidisciplinary teams.

Eppinger *et al.* (1997) and Hanssen (2000) already described the concept of preliminary information. From their work it is known that starting preliminary information increases the risk and can cause iterations, which results in extra cost and delays (Eppinger *et al.*, 1997; Hanssen, 2000). Therefore it is important to release preliminary information in a controlled manner and to communicate changes to the preliminary information as quickly as possible. It is expected that information systems containing Product Data Management (PDM) functionality can be effectively used to control the release of preliminary information and the communication of changes. The focus here is on the type of functionality and not on the system that provides it. Therefore PDM functionality is not necessarily provided by a commercial (off-the-shelf) PDM system, it can also be provided by an in-house developed information system.

1.2 Problem statement

This section describes the problem statement and the reason for my research. According to Hutjes and Van Buuren (1992) a problem statement needs to address three questions:

- What is the phenomenon that is being studied?
- Which conceptual model is used to study this problem?
- What is the research domain?

All three questions are addressed in this section and at the end it is summarised in the problem statement for this research.

Phenomenon

The phenomenon that is studied is the use of PDM functionality in a concurrent product development process where information is exchanged in preliminary form. Where information refers to product definition information, information that contains a description of the product, and not to project information, financial information or any other type of information. The phenomenon can be studied from different angles. In this research concurrent product development processes are studied from a document point of view. The main reason is that product definition information is in most cases still stored in documents, thus starting on preliminary information means starting on preliminary documents that contain the preliminary product definition information. The remainder of this paragraph is used to introduce starting on preliminary information and product data management to the reader.

To explain starting on preliminary (product definition) information, a concurrent development process is compared to a traditional sequential or phased gate process (Cooper, 1993). In a phased gate process the product development process is divided in several phases, for instance conceptual engineering, detail engineering, prototyping, production engineering, and pilot production run, which are separated by so called gates. At these gates it is decided to continue with the next phase or not. If it is decided to continue with the next phase, all the information of the completed phase is released in one large batch to the next phase. Before it is released it is checked. Therefore only stable, consolidated, and proven information is released to the next phase.

In a concurrent development process you work in several phases at the same time. This is only possible if information is released before a phase is completed, such information is considered to be tentative, untested and possibly incorrect. In this research, tentative, untested and possibly incorrect information is called preliminary information and is stored in the form of documents. The preliminary information is exchanged by exchanging preliminary versions of the documents that contain the preliminary information.

Preliminary information is typically released in small batches instead of large batches at the end of a phase (Reinertsen, 1997). Such a small batch of information contains just enough information to start specific activities of the next phase as early as possible. A typical example is a preliminary material specification that is required to start downstream procurement activities as early as possible. In the Construction industry, for example, it is important to order piles as early as possible because they are long lead items. Releasing the material specification at the end of the design process (when it is stable, consolidated and proven) would result in delays because of the long lead-time of the piles.

However, starting on preliminary information increases the complexity of controlling the product development process. Firstly because smaller batches of information need to be monitored in order to check whether they meet their due date. Secondly because there is a chance that preliminary information changes after it is passed to the next phase, after all it concerns tentative, untested and possibly incorrect information. This can lead to iterations in the product development process and thus in extra development cost and delays. Therefore a controlled release and change process is assumed to be essential for starting on preliminary information.

The process of starting on preliminary information could be controlled manually but is also possible to control it using information and communication technology (ICT). In this research the focus is on so called Product Data Management (PDM) functionality. Systems containing PDM functionality are used to manage product definition data through the entire product lifecycle. PDM functionality is commonly used to manage document and document related workflows such as the release and change process. Because starting on preliminary information is considered to be a special type of document release it is expected that PDM functionality is an ‘enabler’ for starting on preliminary information in concurrent product development processes. Although several authors studied PDM, *e.g.* Obank *et al.* (1995), Peltonen *et al.* (1996), Hamer & Lepoeter (1996), Hameri (1998), Sackett *et al.* (1998), Chen & Tsao (1998) and Doblies (1998). None of these authors studied the ‘enabling’ effect of PDM with respect to starting on preliminary information.

Conceptual model

To study the phenomenon one can use a conceptual model that accentuates the aspects that are considered to be important and leaves out details that are considered to be unimportant. In this research it concerns a conceptual model of concurrent product development processes and focuses on the documents in this process that contain product definition information. It can involve analogue as well as digital documents and examples are specifications, datasheets, calculations and drawings.

To study the preliminary release process of documents it is required to introduce the term document lifecycle. Throughout its life a document goes through several stages or statuses. A commonly used document lifecycle in companies consists of the stages ‘in work’, ‘for review’, ‘for approval’ and ‘released’, where each status tells something about the maturity or quality of the document. Traditionally these document lifecycles do not take the preliminary release of documents into account. In this research starting on preliminary documents is considered to be variant of the traditional document release process. Hence starting on preliminary documents can be modelled using the document lifecycle. What is required is the introduction of one or more extra statuses. These statuses represent the moment at which the document can be released preliminarily.

The system that is studied, a concurrent product development process, does not consist of one single document lifecycle but consists of a network of related document lifecycles. If a document is preliminary released this is often the starting point for another document lifecycle resulting in a network of related document lifecycles. PDM functionality fits into this picture because it can be used to model individual document lifecycles as well as networks of document lifecycles. PDM functionality makes it possible to automate or accelerate part of the process. But more important is that it increases the transparency of the product development process, it becomes possible to monitor the product development process continuously. This increases the capability to control the process of starting on preliminary information.

The conceptual model or research model is further elaborated in section 4 and follows the literature review that is presented in section 2 (Product Data Management) and section 3 (Product Development Processes).

Research domain

The research domain involves product development processes that are or could possibly be overlapped by starting on preliminary information. The product development process consists of design and engineering processes that provide the product definition that is required by production. These design and engineering processes are found in repetitive manufacturing, examples are aeroplanes, cars and consumer products, as well as in companies that develop one-of-a-kind products, examples are satellites, industrial plants, tunnels and packaging equipment. For starting on preliminary information the interface with production is also relevant because this is an important interface at which information is preliminarily released. Therefore the focus is on design and engineering processes in product development including the interface with production.

Eppinger *et al.* (1997) introduced two parameters: sensitivity (high/low) and evolution (fast/slow). The combination of extreme values of these parameters results in four extreme situations that can occur in product development processes (this framework is discussed in more detail in section 3.3.3.1). Each extreme situation requires its own approach for overlapping upstream and downstream development processes. In this research the scope is limited to one approach that I refer to as starting on preliminary information. It involves a product development process with low sensitivity and slow evolution. Low sensitivity means that even when changes in the exchanged preliminary information are large their effect on the downstream process is small. Slow evolution means that major changes happen until late into the upstream process, in other words the maturity or quality of the information increases slowly. Low sensitivity and slow evolution are not bound to specific product development processes, products, industries or companies. Eppinger *et al.* (1997) indicate that all four situations can occur in one single company.

Hanssen (2000) also refers to starting on preliminary information in situations of low sensitivity and slow evolution. He concludes that it is only possible to start on preliminary information when it is possible to define beforehand what information should be released preliminary and what the content of that information should be. This is the case when the uncertainty is not too high so that the outcome of the process is more or less known. This is not the case for product development processes that aim for breakthrough innovations. It involves product development processes that aim for improvements or new features in existing products or new products to an existing line of products. The product development process in the One-of-a-Kind business also falls in this category, they use (known) basic products, technologies or processes to create 'new' products that meet the specifications of an individual customer (discussed in more detail in section 3.1.3).

It does not make sense to use expensive ICT technology to support simple processes. Therefore it is expected that a certain complexity is required to make it worthwhile to use PDM in a company. The following factors are considered to influence the complexity of a product development process:

- the complexity of the product,
- the modularity of a product,
- the number of changes (as a result of uncertainty and risk) and
- the geographical spread of the project members (all are discussed in more detail in section 5.4).

Summarising the above it leads to the following **problem statement**:

> *Which PDM functions can be used to control the complexity that is caused by starting on preliminary information in an overlapped product development process and under what conditions do these functions act as an enabler?*

This problem statement is the starting point of my research. The next section describes the research objectives and the deliverables.

1.3 Research objectives and deliverables

The research objectives are the goals that should be achieved by this research. Achieving the goals results in a number of deliverables that are relevant from a business and a scientific point of view. Both the research questions and the deliverables are presented in this section.

The problem statement can be split up in several parts resulting in the research questions:

- *What is preliminary information?*

In order to be able to study starting on preliminary information it is needed to define 'preliminary information'. Because most product development processes still generate information in the form of (electronic) documents 'preliminary information' is defined from a document point of view.

- *How to control starting on preliminary information?*

It is expected that starting on preliminary information increases the control complexity of the product development process, especially in complex product development processes. The complexity requires more communication and co-ordination while the risk of re-work and delays increases. How to control this complexity and the associated risks is explained by describing the process of starting on preliminary information using release and change procedures of documents.

- *What are typical PDM functions?*

PDM is a wide collection of technologies and functions. Moreover, it is often used as a tool to integrate several applications. Therefore it is important for this research to define the typical PDM functions.

- *How does PDM functionality support starting on preliminary information?*

The definition of the PDM functions is used to describe how these can be used to control the complexity caused by starting on preliminary information in the product development process. This is key to understanding the benefits of using PDM for this purpose.

- *Is the use of PDM more efficient and effective than traditional methods?*

Buying commercial PDM systems or developing an in-house PDM system is expensive. Therefore it is only worthwhile to implement such a system if it results in a more efficient and effective process. Therefore controlling the complexity of starting on preliminary information using PDM is compared with more traditional, *i.e.* manual, methods.

The deliverables of this research can be split in two types of deliverables that are relevant from a business and a scientific point of view. From a business point of view the research should result in a description of how companies can use their existing PDM systems to support starting on preliminary information in order to reduce cycle time. While there are many companies that are implementing or have implemented a PDM system (CIMdata, 2000), there are a lot of companies that can potentially benefit from such knowledge.

From a scientific point of view there is one main deliverable, a theory that describes the enabling effect of PDM functionality with respect to starting on preliminary information. But developing this theory also results in three other deliverables:

1. a generic framework to describe the functions of a PDM system.
2. an improved description of document version and document status that contributes to a better method for modelling document lifecycles and document workflows.
3. a conceptual model, *i.e.* research model, of the product development process.

The conceptual model has been used in the RapidPDM project (for more information on this project is referred to Appendix 5) to develop a simulation model to study the effect of PDM on cycle time, cost and quality of product development processes. Only the conceptual model behind the simulation model is a deliverable of this research. The simulation model itself is a combined effort of participants in the RapidPDM project.

1.4 Research methodology

This section gives an overview of the research methodology that is used for this research. First two primitive forms of research are discussed. One of these two approaches is selected and used as an outline for describing the research design.

1.4.1 Primitive forms of research

De Leeuw (1996) describes that there are two primitive forms of research, the *empirical cycle* (De Groot, 1994; De Leeuw, 1996) and the *regulative cycle* (Van Strien, 1986). The empirical cycle is the traditional approach towards scientific research and has been used for a long time. The regulative cycle originated in the previous century in the relatively young (when compared to physics) field of Management Science and Industrial Engineering[1].

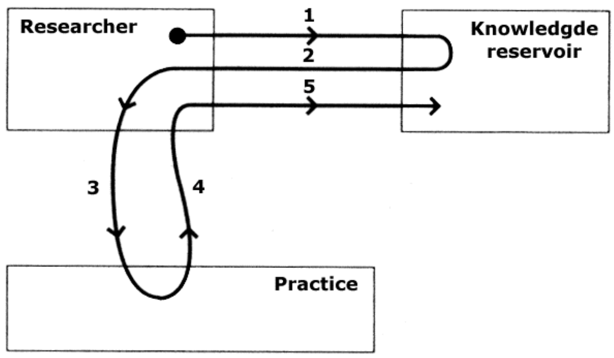


Figure 2 - Empirical Research Cycle

The empirical cycle is discussed in a bit more detail using Figure 2 from De Leeuw (1996). In the empirical cycle the researcher starts with collecting facts from the knowledge reservoir to create a theoretical framework for his research (arrow 1 in Figure 2). Consequently the researcher constructs the hypothesis based on the theoretical framework and derives predictions from these hypotheses (arrow 2 and 3). Next the researcher tests his predictions in an empirical environment by conducting a case study or an experiment, for example (arrow 4). Finally the researcher evaluates the new empirical material and reports his findings that can be added to the knowledge reservoir (arrow 5).

The regulative cycle aims at effective interference in organisations. Therefore it uses a different approach when it comes to practice and the knowledge reservoir. It starts with exploring the problems in an organisation. Once the problems are known the knowledge reservoir is consulted to define a solution for the problems of the organisation. The result of the regulative cycle is a solution, based on scientific knowledge, which is implemented in the organisation. Therefore this type of research is also called design-oriented research.

This research follows the empirical cycle because the main objective is to develop a theory that can be applied in a broad domain. The regulative cycle only results in a solution for one single company; *i.e.* the result is not generalised into a theory for a broader domain. The next section describes the consequences of this choice for the research methodology and sketches the outline of the research design.

[1] Dutch: Bedrijfskundig onderzoek

1.4.2 Research design

The research design is described using the model of De Leeuw (Figure 2). The design starts at the end of the empirical cycle by looking at the result, because it is the main determinant for the research design. The result of this research is a theory, as mentioned in the previous section, tested by empirical observations. There are several methodological approaches to test a theory such as an experiment in a laboratory or a survey. But in this research the case study approach has been selected. The reason behind this choice is threefold:

- The subject of study is not well known and not all variables that play a role can be identified beforehand. Hence a more holistic approach is required in which a lot of variables are taken into account.
- It is not only of interest to know 'if' PDM is an enabler for starting on preliminary information, also the conditions in which PDM acts as an enabler are of interest. Hence more in-depth research is required.
- In my consulting activities I had an excellent opportunity to look 'into the kitchen' of various companies.

For a long time, however, case studies were not widely accepted as a methodological approach for conducting scientific research. It was not considered to be a rigorous type of research that would lead to scientific results (Den Hertog & Van Sluijs, 1995). However, several authors, amongst others Straus & Corbin (1990), Miles & Huberman (1984) and Yin (1994), further developed the case study approach into a more mature research approach that is widely accepted today. Case studies can be used for different types of research and not only for exploration or developing a theory. It can also be used for testing or validating theories and eliminating rivalling theories.

In this research the case study is used to test or validate the theory on PDM functionality and starting on preliminary information. Now the outline of the research design is presented, a more detailed description is provided using the five steps of the empirical cycle.

Step 1 - Establish theoretical framework

The first step concerns the development of a theory on the enabling effect of PDM functionality with respect to starting on preliminary information in overlapped product development processes. This theory is based on three main sources.

The first source is Product Data Management theory with a focus on identifying and defining typical PDM functions. It is assumed that once the typical PDM functions are defined it becomes possible to develop a theory on how these functions support starting on preliminary information. The PDM functionality is not necessarily provided by a commercial (off-the-shelf) PDM system, it can also be provided by an in-house developed information system. The goal of studying PDM literature is to develop a PDM function model that defines the typical PDM functions (chapter 2).

The second source is Concurrent Engineering theory with a focus on starting on preliminary information in overlapped product development processes. Concurrent Engineering is studied from a document point of view. Therefore document release and change processes play an important role here. These processes are common in product development and are described in standards such as ISO 9001 for Quality Management Systems (NNI, 1999) and ISO 10007 or MIL-HDBK-61 for Configuration Management (DoD, 1997). Because there is relatively little literature available on the preliminary release of documents additional attention is paid to modelling the preliminary release process of documents (chapter 3). This model helps to better understand the problems that are caused by starting on preliminary information.

The linking pin between the first two sources of information is the theory on Business Process Design (Davenport and Short, 1990) (Hammer, 1990) (Romney, 1995). This theory states that there is a recursive relationship between Information Technology and Process Design. It means that the design of the process determines which information technology is required but on the other hand information technology also determines the design of the process. Or in other words new process designs are ‘enabled’ by applying information technology. The goal of studying this literature is to better understand the term ‘enabler’ (section 5.1).

Step 2 & 3 - Formulate hypotheses and make predictions
The second step concerns the development of a number of hypotheses. The hypotheses are used to validate specific parts in the developed theory. The theory itself is based on literature research and explanation building and requires validation in an empirical situation. The empirical validation is used to validate certain parts in the theory that are considered to be essential for the reasoning that is used in the theory. In other words without the validation of these parts the theory will not hold.
The hypotheses are in fact predictions that follow from the theory. The predictions are tested in a number of case studies to generate empirical data. The data from the case studies is used to improve the validity of the theory. The result is a theory that is based on literature, explanation building and is backed by empirical validation.

Step 4 - Empirical research to support hypotheses
The fourth step concerns the collection of empirical data. For collecting empirical data the case study approach is used, as mentioned in the introduction of this section. In the following the outline of the case study design is discussed, but the details of the case study design are saved until section 6.
The goal is to study which PDM functions are used or developed by companies that start on preliminary information and under what conditions do these functions act as an enabler. First of all this requires that it is defined under which conditions PDM functions act as an enabler, this is already covered by the first step in which the theoretical framework is established. Secondly companies should be identified that start on preliminary information under the specified conditions. An efficient way to identify such companies is to consult experts in the field and ask them if they know companies that meet the specified criteria. Of course the identified companies should be visited to check if they meet the criteria, if they do they can be included in the research.
However, how many companies should be involved in the case study? Because case studies rely on *analytical generalisation* instead of *statistical generalisation* it is not required to have a large sample, *i.e.* a lot of companies, that is representative for the whole population (Yin, 1994; Hutjes and Van Buuren, 1992; Swanborn, 1996). The aim of analytical generalisation is to develop a theory and not to statistically proof for instance a relation between two phenomena. Therefore authors such as Yin (1994), Hutjes and Van Buuren (1992), Swanborn (1996), Straus and Corbin (1990), Miles and Huberman (1984) argue that only a small number of cases is required to achieve analytical generalisation.
For this research it was decided to do more than one case study because this increases the possibility to generalise the results (Yin, 1994; Hutjes and Van Buuren, 1992; Swanborn, 1996). However, because of time and money constraints it was decided to conduct no more than two case studies. To increase the likelihood that it is possible to generalise the results it was decided to use the technique of maximum variation (Hutjes and Van Buuren, 1992). In such a set-up two companies with different characteristics are selected, for instance companies in different industries with different products and different markets. If the case studies have similar outcomes the company characteristics do not influence the outcome. Such

an outcome increases the likelihood that it is possible to generalise the results, because the theory is tested in two totally different companies.
Moreover, Yin (1994) described several quality criteria for a case study design. If the case study design meets these quality criteria it leads to more reliable results. The application of the quality criteria to this research is saved for section 6 in which the details of the case study design are discussed.

Step 5 - Evaluate findings and draw conclusions
The final phase of the empirical cycle is analysing the results of the case studies. The conclusions are formulated by reviewing all the facts that follow from the case studies. The main technique that is used here is explanation building in which empirical facts from the case studies are used. This is followed by a discussion on the likeliness that the results can be generalised to companies outside the research domain. From this discussion follow recommendations for further research.

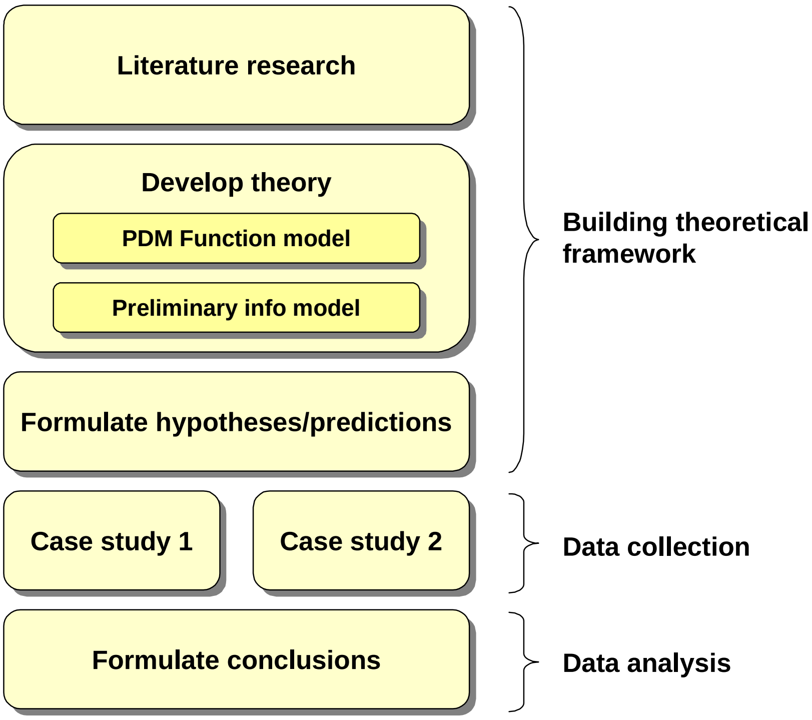


Figure 3 - Overview of Research Design

The research design is summarised in Figure 3 and shows the different steps that have been discussed in this section.

1.5 Dissertation overview

The remainder of this dissertation consists of 6 chapters and can be roughly divided in three parts, as was shown in Figure 3. In part one (covered by chapter 2, 3, 4, and 5) the theoretical framework is build; it describes the literature review that results in a theory that is tested in the case studies. In part two (covered by chapter 6) the collection of the empirical data is described; it describes the results of the two case studies. In part three (covered by chapter 7) the data from the case study is analysed resulting in the conclusions of this research. The following list describes each chapter in a bit more detail.

Chapter 2 presents a generic framework to describe PDM functionality as well as the benefits of PDM; later chapters will refer to the PDM functions that are defined by this framework.

Chapter 3 presents an overview of product development processes and focuses on Concurrent Engineering and exchanging preliminary information to overlap phases in the product development process.

Chapter 4 presents a conceptual model of the product development process from a document point of view; the model is used to describe starting on preliminary information in more detail.

Chapter 5 presents how PDM functionality can be used to control the complexity that is caused by starting on preliminary information; this chapter results in the final hypotheses that are tested in the case studies to collect empirical data to support my theory.

Chapter 6 presents the case study design and the results of the two case studies that have been conducted; it involves a case study in the aerospace and the process industry.

Chapter 7 presents the conclusions of this research that are based on the findings of the two case studies.

2 Product Data Management

In this chapter the concept of Product Data Management (PDM) is presented as well as the technologies that are used to realise this concept. The chapter is used to indicate what is regarded to be PDM functionality and what is not. If it is clear what can be regarded as PDM functionality it becomes possible to speak about the benefits of PDM realised by these functions.

2.1 Definition of PDM

Product Data Management (PDM) is known under several names (McIntosh, 1995). Examples are Technical Information Management (TIM), Engineering Document Management (EDM), Product Information Management (PIM), and Engineering Data Management (EDM). The latest term that is introduced is cPDM, which stands for Collaborative Product Data Management and was introduced by CIMdata. Gartner Group refers to this phenomenon as CPC that stands for Collaborative Product Commerce. CPC and cPDM both represent a shift in focus from individual companies to the supply chain. However, in this thesis I stick to the term PDM and give the definition of PDM in the context of this research.

Several PDM definitions can be found in literature (Helms, 2000). It involves definitions from Bret (1994), CIMdata (1995), De Kok *et al.* (1995), McIntosh (1995), Gartner Group (1997, 1998), Goossenaerts & Pels (1998) and Breuls (1997, 1999). All definitions have in common that PDM concerns the management of data across the product life cycle; that is from its inception, development, test, and manufacture, through its ultimate demise. However, some definitions define more clearly which data is managed than the other definitions. And some consider PDM to be a tool whilst others consider it to be a discipline or business practice.

In this thesis the RapidPDM definition of PDM is used Helms (2000):
‘PDM is the discipline of making the right product and process related data available and accessible to the right parties at the right time in the product lifecycle in order to support all business processes that create and/or use this data’.

The reason behind this choice is that this definition contains the aspect on which all authors agree: PDM manages product data across the product lifecycle. Moreover, this definition underscores that PDM is not just a software package but also a discipline, a company can for example manage document versions without using a PDM system. This is in line with the statement in chapter 1 that PDM functionality is not only found in PDM systems. Furthermore, the definition also includes process data to show that also the process should be managed in which data is created and/or used. This is important because in this research the capability of PDM to support all business processes that create or use this data plays an important role. In this research it is of interest how it supports the processes that create and/or use preliminary information.

2.2 Typology of PDM functions

This section gives a short overview of typologies of PDM functions from literature. Based on one of these typologies a PDM function model has been developed within the framework of the RapidPDM project (for details on the project see Appendix 5). This PDM function model will be used here to describe what is considered to be PDM functionality. These definitions are used later in this thesis to describe how PDM can support Concurrent Engineering.

2.2.1 Existing typologies

Several authors described PDM functions or function models. These include models by CIMdata (1995), De Kok *et al.* (1995), McIntosh (1995), Hamer and Lepoeter (1996), OMG (1996), Breuls (1999), and Sackett *et al.* (1998). These models are described and compared in Helms (2000). The comparison leads to the conclusion that the typologies are not completely different and that there is a large degree of overlap.

By comparing all the different typologies in literature Helms (2000) concluded that because of the overlap there is at least consensus about what is PDM functionality and what is not. The functions that were reported by more than one author include:

- Vault Management
- Document Management
- Change Management
- Configuration Management
- Workflow/Process Management
- Product Structure Management
- Retrieval/Access Management

However, the typologies seem to be arbitrary because most authors do not clearly describe how they came to their typology, except for one author, Breuls (1997, 1999). He defines a sound basis for his typology based on the objects that need to be managed by PDM and the information that needs to be managed about these objects. The result is a generic framework that is derived from the needs of a company while much of the other typologies are derived from what PDM systems offer. The approach from Breuls is considered to be a much more fundamental; therefore his approach has been used for developing a PDM function model. I developed this PDM function model within the framework of the RapidPDM project.

2.2.2 PDM Function model

Breuls (1999) bases his model on the information that should be managed by a PDM system. First of all he distinguishes several types of (business) objects that a company should store information about. Secondly he distinguishes several types of information that should be stored about these objects. The types of objects that are distinguished are as follows:

- Activities
- Products
- Resources
- Documents

Firstly, companies need to store information about their product development process, therefore an activity object type should be included. Secondly, a company requires resources, *e.g.* employees and tools to execute these processes. Therefore, a resource object type should be included. Third, the main outcome of the product development is the definition of the product. Therefore, a product object type should also be included. Finally, a product is described using documents and therefore a document object type should be included.

It should be noted that a document is a special object type when compared to the others because documents can contain information about the other objects. For example, a drawing is a type of document that can contain information about a product, *i.e.* its geometry.

The three types of information that are defined by Breuls (1999) include content *information, relationship information* and *life-cycle information*. The types of information are explained using an example of a company that produces luminaries (see Table 1).

Table 1 - Object and information types recognised by Breuls (1997)

	Object data	**Object environmental data**	
Object	Content Information	Relationship Information	Life-Cycle Information
Product: Luminaries	• CAD drawing • Customer specific adaptations	• Bill-of-material • Order and order line identification	• Version 1.0 • Status: development • Owner: prod. mngr.
Activity: Luminaries assembly	• Assembly instructions • Actual activity lead time	• Actual input docs and products • Order identification	• Version 2.0 • Status: completed • Executed by: B
Resource: project leader Z	• Performance description • Location information	• Current subordinates in project P • Current projects	• Authorisation level • Hours spent on project P: 100
Document: Marketing brochure	• Brochure summary • Brochure target group description	• Brochure sub document structure • Related products	• Version 6.0 • Status: Published • Changed: 1998-12-05

According to Breuls (1999) a PDM system should manage these three types of information. This results in the definition of the following three main functions of a PDM system:

- *Object Repository Management*: The collection of activities whose aim is the optimal storage, retrieval and maintenance of data. This data may be stored in electronic format as well as on conventional paper or microfilm archives.
- *Object Structure Management*: The collection of activities aimed at managing relationships between objects. It involves managing object structures such as a product structure or a work breakdown structure.
- *Object Life Cycle Management*: The collection of activities aimed at providing guidance and supporting creating, using, changing and removing objects from the viewpoint of managing related information flows.

These functions are used as the basis for the PDM function model in this research. The function model of Breuls is extended by also taking the type objects into account. The reason is that the three main functions can be different for the different types of objects. For example, the functionality for managing a product structure is different from functionality for managing a document structure. Therefore a PDM function model has been developed that takes the types of object as well as the types of information into account. In other words, one can define functions that focus on the business objects as well as functions that focus on the information types.

These two sets of PDM functions are orthogonal; hence a PDM function matrix or model can be created. This is shown in the following Figure 4. On the vertical axis the functions for managing the different types of information are shown while on the horizontal axis the functions for managing the different types of objects are shown. By combining those two axes the 9 main PDM functions are created (the shaded areas in Figure 4).

Figure 4 - PDM Function model

It should be noted that there are three object oriented PDM functions instead of four (initially Breuls defined activity, resource, product, and document). This is because the ‘activity’ and ‘resource’ object have been combined into one object, *i.e.* ‘process/project’. Resource management is not considered to be the focus of PDM. Therefore it has been decided to combine it with the ‘activity’ object into the ‘process/project’ object. Managing resources and activity are typical project management tasks.

Section 2.2.4 will discuss the nine PDM functions in more detail. However, only the main functions are addressed and not every single function or feature, just enough for the reader to understand what PDM is all about. But first a glossary of terms is introduced. A clear understanding of these terms is essential because they are used throughout this thesis and especially in the section that describes the nine PDM functions.

2.2.3 Glossary of terms

This section presents an overview of the terms that are used throughout this thesis. A lot of these terms are related to database and object oriented terminology; this is not surprising because the 'heart' of a PDM system is a database. However, in the PDM industry sometimes a slightly different definition is used. Therefore, it is important that the reader becomes familiar with these terms before reading the remainder of this thesis[2].

Objects, object classes and attributes

An object represents a 'thing' in a company about which you want to store information in a PDM system. Each single product, part, customer, project, procedure, manual etc. of the company can be represented by an object in the PDM system. An **object class** represents a collection of objects that have similar properties. Object classes can be organised in generalisation hierarchies resulting in **super classes** and **sub classes**. An example of a super class is a 'person', example of subclasses belonging to the super class 'person' are 'employees' and 'clients'. Such a structure of super classes and sub classes results in a **class hierarchy**.

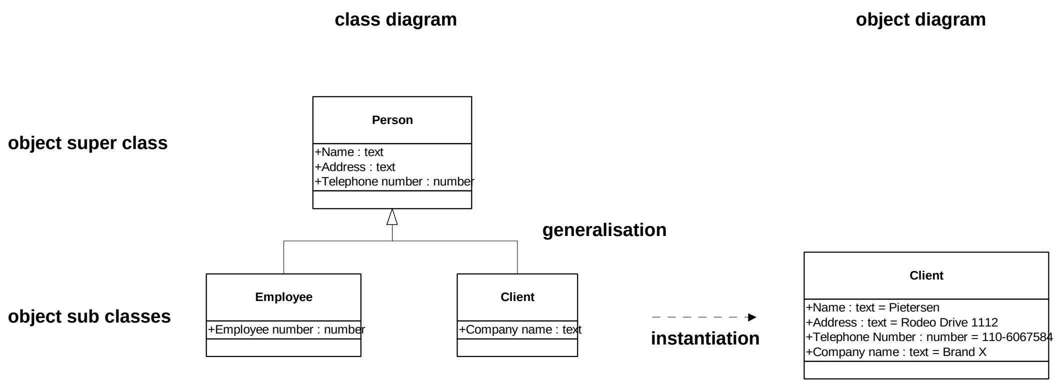


Figure 5 - Example of instantiation of an object from an object class

The properties of an object class are described by **attributes**; each object class has its own set of attributes. Examples of attributes for the super class 'person' are 'title', 'name', 'address', and 'telephone number'. Sub classes inherit the properties of their super class, but each sub class will also have additional properties describing the specific properties of that sub class.
Object classes, super classes and sub classes only describe collections of objects, *i.e.* meta-model. An example is the object class 'person' that describes a collection of persons with the same properties. One particular person called 'Pietersen' is called an **instantiation of an object class** or in short an **object**. Other objects may exist in the database that represent the persons called 'Janssen' and 'Henderson' The names of the persons 'Pietersen', 'Janssen', and 'Henderson' are examples of **attribute values** of the attribute 'name'. When creating a new instantiation of the object class 'client' in a PDM system; the attribute values of that particular client have to be entered. The creation of objects fills the PDM system with information about these objects (see Figure 5).

[2] Note: The terminology used in this section is derived from theory on object-oriented thinking. For more information on this subject refer to Rumbaugh and Booch (1996).

Relationship classes and relationships

Objects can be linked using relationships. Like object classes there are **relationship classes.** Examples of relationship classes are ‘is part of’ and ‘is derived from’, also called types of relationships. The ‘is part of’ relationship indicates that there is a parent-child relationship between the linked objects, while the ‘is derived from’ relationship indicates that one object is derived from the other object. In a PDM system it is often possible to constrain the relationships that can be applied to certain objects. For example, ‘is part of’ relationships can only be applied to objects that belong to the object class ‘product’.

Special attention should be paid to the ‘is a successor/version of’ relationship class, because it is used for **object versioning**. Object versions are ‘meaningful snapshots of an object over time’ (discussed in more detail in section 3.2.3). Each object version is therefore a successor of the previous object version. An individual object represents each object version in the PDM system. Linking objects using the ‘is a successor of’ relationship results in a chain of object versions. When using the notion of object versions, the object can still exist carrying the attribute values that are common for all object versions. The versions are then linked to the object by relationships.

Meta-data, vault, and repository

In the database world the definition of object classes and attributes are called meta-data, which refers to the term meta-model meaning model of a model. Therefore meta-data is data about data. In the database world the instantiations of object classes and attributes are called data and the object classes and attributes are therefore data about data or meta-data. In the PDM world the instantiations of object classes and attributes are called meta-data because it contains data about the (computer) files that are stored in the PDM system. For object classes and attributes there is no definition in the PDM world, but in fact it is meta-meta-data.

The (computer) files that are managed in a PDM system are stored in so-called **vaults**. Files that are stored in a vault can only be accessed via the PDM system. Storing a file in the vault is called a **check-in**. If the file is retrieved for modification it is (temporarily) **checked-out**. After making the modification the file is checked-in again. Often it is possible to specify what should happen at check-in, *i.e.* overwriting the old file or creating a new version.

In the PDM world the place where the meta-data and the vaults are stored is referred to as the **repository**. Once again this is different from the database world because they use the term repository for the place where meta-data is stored. In the PDM world there is no name for storing the meta-meta-data.

Document
In this thesis the terms document, abstract document, document type, and document version are used. Using the terminology presented in this section they are all examples of object classes. The term **abstract document** is introduced to refer to the piece of information that must be delivered to specify a product (*i.e.* its form, fit, and function) or a process. An abstract document object has attributes for invariant properties like id, title, and author (see Figure 6). During the life of a document its content may change. Every relevant snapshot of a document is stored as a document version object, which has attributes for at least the version number and the location of the corresponding file.

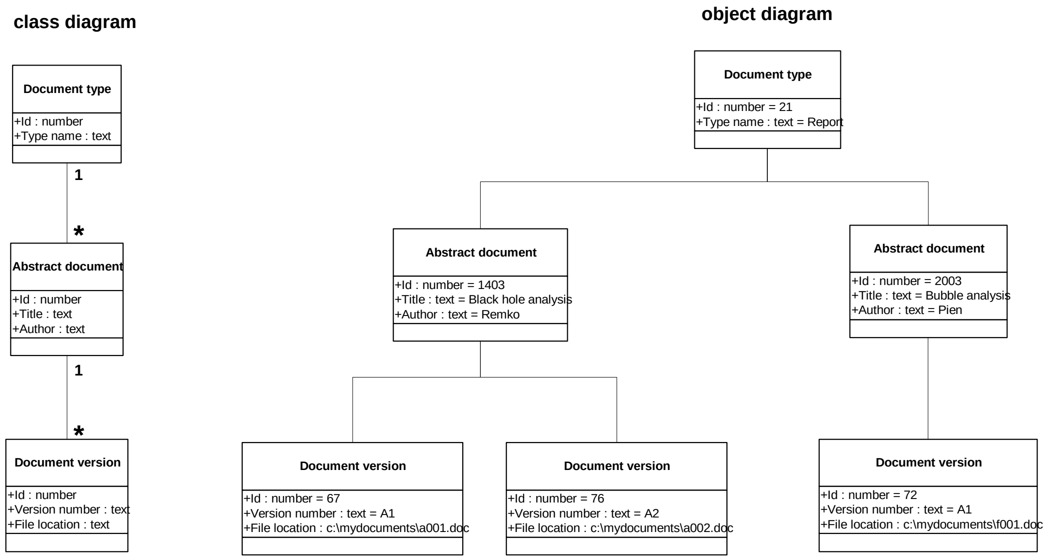


Figure 6 - Relation between document type, abstract document and document version

An abstract document is always of a certain **document type**, examples of document types are report, memo, datasheet, drawing, specification, and project plan. Document types can have templates and lifecycles. In this thesis the term document is used interchangeably with abstract document but its interpretation should be clear from the context. A document version is always a document version and no other terminology is used to refer to a document version. It should be noted that in practice the terms document and document version are often mixed up. When somebody uses the term **document** he actually refers to a specific version of a document. Still the term document is used because people automatically refer to the latest or effective **document version**. Therefore they will refer to the document: material specification of part X, instead of referring to the document version: version Y of the material specification of part X.

2.2.4 PDM Functions

The three functions Object Repository Management, Object Structure Management and Object Life-cycle Management are used to describe the generic functions of a PDM system. After that the specific functionality for managing products, documents and processes/projects is described here in more detail.

2.2.4.1 Generic PDM functions

Repository Management

It was already mentioned in the previous section that the basis of a PDM system is an object-oriented database. Hence one can define object super and sub classes, relationship classes and attributes to set up the PDM system. Typically the following object classes are defined in a PDM system: document, product, and project (see also section 2.2.1). By defining the attributes for each object class the user defines what information he wants to manage about the objects.

The user fills the PDM system with information about documents, products and projects of the company. The attribute values of the objects enable the user to retrieve information from the PDM system. For example, a user can query (*i.e.* search) the repository for all objects that are created by the same person or for the same customer.

Structure Management

Besides storing information about single objects it is also possible to define relationships between objects, this results in object structures. For instance, product objects can be connected to another object using a “is part of” relationship. An attribute of this relationship could be the quantity, *i.e.* how many pieces of a component are used in the assembly. In this manner it is possible to model product structures and bill of materials using a PDM system.

Besides “is part of” relationships the user can also define other types of relationships, for instance “is derived from”. By creating relationships between objects the user creates a network of objects. These networks also contain useful information for the user. It enables the user to find information that is related to a specific object, the components in an assembly for example.

Lifecycle Management

Also the time aspect of an object should be taken into account. Life cycle management enables to define the necessary life cycle steps with corresponding statuses and serves and helps with the use of workflow functionality to push objects through their life cycles. There are rules about how an object can go from one state to another. These state transitions are triggered by events in a workflow.

Finally, over time there can be several versions of an object or an object structure. These versions are snapshots of the object over time. Together the object versions tell something about the history or evolution of an object. In the previous section was mentioned that object versions are modelled as “ is a successor of” relationship between objects. This suggests that this should be part of Structure Management functionality. However, because of the time aspect it is considered to be part of the Lifecycle Management functionality.

2.2.4.2 Document Management

The general principles (*i.e.* Repository, Structure and Lifecycle Management) can be translated into specific functions for managing the information with respect to (abstract) documents and document versions. A PDM system contains an object class (abstract) document, document version, and document type. For each document type a different set of attribute types should be defined in the PDM system. If the user adds a new document to the PDM system, the system creates a new object with a unique number and the user has to fill in the meta-data. Although the PDM system assigns a unique number to each document in the PDM system, it should also provide functionality to support the different numbering schemes that are in use in product development organisations. N.B. Only documents have unique number, document versions have identical identification number but different version numbers.

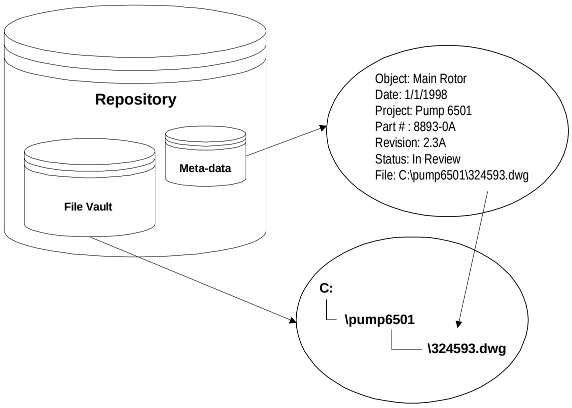


Figure 7 – Relation between object and file

Besides managing information about a document, companies also want to manage the corresponding file(s). These files are always linked to the document version object and not to the (abstract) document object. Files are stored in the vault where they are under the control of the PDM system (see Figure 7). In case a document version is not electronically available a PDM systems offers capabilities to scan hard copy document versions. The PDM system manages the relation between the file in the vault and the document version object. Hence the user does not need to worry if the stores the files on the right place. The user can retrieve a file by searching for the attribute values of the (abstract) document object. If the document is found the user can edit or view the version history of the document and select the document version he needs. After selecting the right document version he can view the corresponding file. The user does not need to worry about the format of the file. For example, if a user does not have AutoCAD installed, the PDM system will start a viewer to view the drawing.

A PDM system should also support the ability to create document structures, *i.e.* define relationships between documents and document versions. Several types of relationships can be distinguished. First of all there can be hierarchical and lateral relationships. Second there can be internal and external relationships.
Hierarchical relationships are used to define that for instance a manual is composed of a user manual and a maintenance manual that are managed individually or to define that a product dossier consists of drawings, changes, and calculations. Lateral or logical relationships, on the other hand, are relationships between documents on an equal level. An example here is a particular version of a calculation that belongs to particular version of a drawing.

External relationships are relations between two or more documents that are created by the PDM system. The mentioned examples of hierarchical and lateral relationships are both external relationships. Internal relationships, on the other hand, are relationships inside a file and are not (necessarily) created by the PDM system. An example is an X-refs in an AutoCAD file. The X-refs refer to other AutoCAD files that for instance represent the mono-parts of an assembly. A PDM system should be able to manage or at least be able to recognise these relationships.
Document structures can be helpful in finding related information; it provides the user the possibility to graphically browse through the structure. By following these relationships the user can find the information he needs. These structures can also be used to define the impact of an engineering change. If a document needs to be changed because of an engineering change the relationships with other documents are an indication of other documents that are possibly also affected by the change.

Finally, a PDM system should support the lifecycle of a document, which means that it should support document status and version control. Status control is used to define the states of a document during its lifecycle; this is for example used to support the release process of a document. For each state it is defined who is authorised to access, modify or delete the document.
Version control is used to guide the creation of several versions of a document. By managing document versions the user has access to the history of a document. Document versions also have a lifecycle, which it inherits from the document it belongs to.
Status and version control is not only applied to individual documents but also to document structures. This makes it possible to see which documents have been parts of a document structure over time. Moreover, it is possible to store a specific document structure for later reference, which is also called a document baseline. Examples are an as-designed and an as-built document baseline.

Summarised, the main document management functions and sub-functions are:

- Document Management
    - Document Repository Management
        - Vault management
        - Document identification & numbering
        - Document viewing and mark-up
        - Scanning and imaging
    - Document Structure Management
        - Document structure modelling
        - Document structure viewing
        - Document change impact analysis
    - Document Lifecycle Management
        - Document version control
        - Document status and release control
        - Document baselines

2.2.4.3 Product Management

The general principles previously described can also be translated into specific functions for managing the information with respect to products. The PDM system contains an object class (abstract) product, product version and product type. Examples of product types are spare part kits, end products, and raw material or buy items. For each product type different sets of attributes can be defined in the PDM system. Often this kind of information is also stored in other databases, for instance the Enterprise Resource Planning (ERP) system. However, this information is often not easily accessible for engineers. Moreover, it does not always contain the information in which an engineer is interested.

Important functionality on the repository layer is product identification and classification. In practice companies have different procedures for their numbering that should be supported. Moreover, it is not uncommon to reserve ranges of numbers for a specific purpose. Often companies also use intelligent numbers, this means that specific information with respect to the part is coded in the identification number, *i.e.* part number. PDM systems should support the automatic generation of these 'intelligent' numbers.

Classification is important for re-use of parts and assemblies. Classifying an item means that it is assigned to a class of products, for instance pumps. Each class is characterised by its own set of parameters, for a pump this might be manufacturer, flange diameter, input pressure, and output pressure. This classification makes it possible to look for all pumps with a specific flange diameter[3].

An important function is the ability to create product structures. It concerns the breakdown of the products into assemblies and components and/or the breakdown of the product into its functions and sub-functions. A typical example of a product structure is the (engineer/manufacturing) Bill of Material. By modelling the product the engineer in fact configures the product. If a product contains a lot of standard parts a product could also be configured by selecting the required modules, this could also be implemented as setting parameters to configure the product. Such a function is also called a product configurator.

In addition to hierarchical relationships it should also be possible to create lateral relationships. An example of a lateral relationship is a relationship that indicates that two parts are interchangeable. Moreover, it should also be possible to link documents or document versions to a product (structure). Linking drawings and calculations to products makes it easier to retrieve these product-related documents. It enables the user to search in the PDM system for all the documents or document versions related to a specific version of a component.

Of course it should also be possible to view and browse or navigate through the product structure. This means that the product structure is graphically shown on the screen and that the user can follow the relationships between the parts and assemblies. This enables the user to browse through the products and/or documents that he needs instead of defining queries.

One step further is viewing a 3D model of the product. In this case the user can browse through the 3D model. By double clicking on an assembly, for instance, he can see the related information or switch to the Bill of Material of that assembly. The product structure can also be used for some calculations. If for example the cost price and weight is known for each individual part it is possible to calculate the weight and the cost price of the total product.

[3] For further reading on classification the reader is referred to literature on Group Technology.

Finally, the product structure can also be used to determine the impact of a change. It supports finding components and assemblies that are related to the affected components as well as the documents that are related to these components and assemblies. In the case of a change it is important that the user can lock or flag the affected parts and documents so that other users do not use this information.

Over time the product configuration changes, new versions of components and assemblies emerge and components are deleted from the product configuration or replaced by other components. This requires version control on the component level as well as on the product structure level.

Sometimes it is required to make snapshots of (or freeze) the complete configuration and its related documentation. Examples here are an as-designed or an as-built baseline and often these baselines coincide with a milestone in the project. The baseline then represents the information that is the basis for the next phase in the project.

Status control is used to define the states of a component or assembly during its lifecycle; this is for example used to support the release process of a component or assembly. For each state it is defined who is authorised to access, move or delete the parts or assemblies and the related documents. Examples of statuses of an assembly include Released for Purchasing, Released for Prototyping and Released for Production. Closely related to this function is the effectivity of a component or assembly. An example is that a new component is effective in a product configuration if it has a specific date or serial number.

Summarised, the main product management functions and sub-functions are:

- Product Management
    - Product Repository Management
        - Product identification
        - Product catalogue/classification
    - Product Structure Management
        - Product structure modelling
        - Product configuration
        - Product structure viewing
        - Product structure calculations
        - Product change impact analysis
    - Product Lifecycle Management
        - Product version control
        - Product status and release control
        - Product baselines

2.2.4.4 Process/Project Management

Finally, the general principles can also be translated into specific functions for managing the information with respect to processes/projects. The repository function involves the storage of information with respect to projects, such as activities and work packages. The identification of work packages is an important function because often companies have to use the numbering method of their customer. These numbers should be generated on request when a new work package is created.

Another function on the repository management level is traceability. It should store all the process information that is required for setting up an audit trail. Such a trail is important in the context of ISO 9001 or product reliability claims.

Structure management involves managing the project or the work breakdown structure. It involves the decomposition of a project into work packages and tasks that should be conducted. Modelling such a work breakdown structure is also one of the functions of a PDM system. It should be possible to link deliverables, *i.e.* documents or sets of documents, to a work package. By means of a graphical presentation of the work breakdown structure the user can then browse or navigate through it and find documents related to a particular work package.
A related structure is the organisation breakdown; it shows which employees are involved and their position in the project. Such a breakdown is often used in a PDM system to determine the authorisation rules. Depending on somebody's position he has or does not have rights to view, modify or delete information.

The lifecycle management aspect of process/project management involves work planning and workflow support. By means of a work breakdown structure all the work packages and corresponding deliverables can be identified. In order to make sure that all the planned work is completed on time it should be possible to track the deliverables (*i.e.* documents). This means that the user can follow the progress of the project by looking at the progress of the individual deliverables. If deliverables are behind schedule the user can check the corresponding workflow for the status of that specific deliverable. It might be on hold or waiting for input from somebody else. When a deliverable is near or over its due date the system automatically generates reminders/notifications.
With workflow the user can model processes as a number of related events. An example of an event is the sign-off of a document; an authorised person has to put his (electronic) signature on the document. The event is triggered when a document is offered for sign-off, the system then sends the request to the mailboxes of the authorised persons that need to sign-off the document. They open the request, read the document and then sign-off the document by clicking on a sign-off button. The document now receives a higher status and employees are notified about this change in status. Workflows typically support the release and change process of deliverables. However, also other processes can be supported by workflow. An example here is handling queries from the construction site.
Workflows can also have versions; the change procedure can for instance be replaced by a new version. An example is where all the documents that are already in the change process should be completed according the old procedure, while the new documents follow the new change procedure.

Summarised, the main process/project management functions and sub-functions are:
- Process Management
    - Process Repository Management
        - Process identification
        - Process tracking (audit trail)
    - Process Structure Management
        - Work breakdown structure modelling
        - Organisation breakdown structure modelling
        - Structure viewing/browsing
    - Process Lifecycle Management
        - Work planning
        - Work tracking
        - Workflow management

2.2.5 Validation of the function model

The function model presented describes PDM functionality from a functional or user point of view. It will be used to define which PDM functions support Concurrent Engineering, especially starting on preliminary information. Therefore, it is important that the quality or practical usefulness is tested. According to ’t Hart (1997) the validity of such a typology of IT-functions depends on its completeness, accurateness and distinctive power.

1. Completeness

If the PDM function model is incomplete it would have omissions, in other words functions of commercially available PDM systems cannot be mapped on the function model. The problem is, however, that there is no exact definition of PDM functions. The function model itself is an attempt to define the scope of PDM. Discussions in the RapidPDM project, in which two PDM vendors participated as well as PDM consultants, proved that the function model is fairly complete. At that time the participants were not able to define a PDM function that could not be classified.

Another method to check the completeness is to match the function models that are found in literature against the RapidPDM function model. For this match is referred to Helms (2000), it shows that the function model does not suffer from major omissions.

2. Accurateness

According to ‘t Hart (1997) an indicator for both completeness and accurateness is the amount of anomalies (*i.e.* functions of commercially available PDM systems that can not be typified effectively). As was mentioned in the paragraph on completeness, discussions in the RapidPDM project showed that the amount of anomalies is low. There were no functions found that could not be classified in the function model. On the other hand the function model is not capable of showing the differences in detail as they are found in practice. An example is the fact that some PDM systems can define the authorisation for a single document while other PDM systems can only define the authorisation on document type level. This can be an important aspect for a company in the Defence industry where classified documents are used very often. However, this kind of detail is not required here and therefore the accurateness is considered to be sufficient.

3. Distinctive power

The completeness aspect refers to the degree to which the function model covers the total width and breadth of the variety in commercially available PDM systems. The accurateness refers to the question of whether distinctions made within the total width and breadth may be considered correct. The distinctive power of the function model refers to the depth, or degree of detail, to which the function model is capable of recognising differences between PDM functions in commercially available PDM systems. Today the function model is still capable of showing the differences between commercially available PDM systems. Not every PDM system supports the Product Structure Management and Product Life Cycle Management function for instance. Even more PDM systems have a rather limited support of all the Process/Project management functions. However, according to the Gartner Group the PDM market is reaching its maturity, which means that the differences between the PDM systems have become less obvious then in the early 1990s.

2.3 *Architecture of PDM systems*

There is no such thing as a standard architecture for PDM systems. It depends on the situation and the needs of the company as to what kind of architecture will be realised. However, one can always recognise the basic building blocks of an architecture that are briefly discussed in this section.

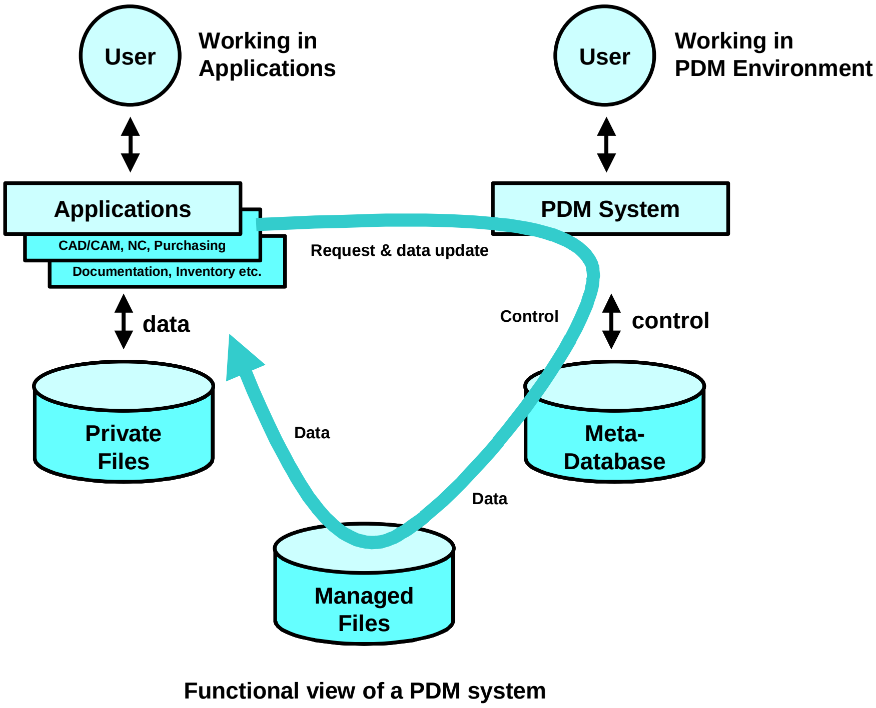


Figure 8 – Principle of a PDM system

Figure 8 gives a simple overview of the architecture a PDM system (CIMdata, 1997). On the left hand side it shows that employees use different applications to create files and that they can store these (private) files somewhere on a file server. On the right hand side it shows the PDM system, consisting of a meta-database and a vault for managing the files. The user can put files under the control of the PDM system, which checks the file into the vault and some meta-data about the file is entered by the user.

In practical terms there are two ways to access files in the PDM system. The first is via the PDM client; this is PDM software that is installed on the user's computer. Secondly the user can access the file using the application that was used to create the file. This application will connect to the PDM system and will request a 'check-out' of the file. However, this requires some integration between the user application and the PDM system.

The level of integration between the user application and the PDM system can vary. The minimum level of integration that is required is that the PDM system knows which application is used to create the file and that it can launch the application with the file the user wants to edit. In the case of a higher level of integration several PDM functions can be started from within the user application, for instance creating a new version of a file in the PDM system.

Most PDM systems today are still based on the client-server model. This means that there is a PDM server that manages the meta-data and the vault and that there are several clients through which the data can be accessed (see Figure 9). The user has to start the PDM client software and log on to the PDM system before he can access the data stored on the PDM server.

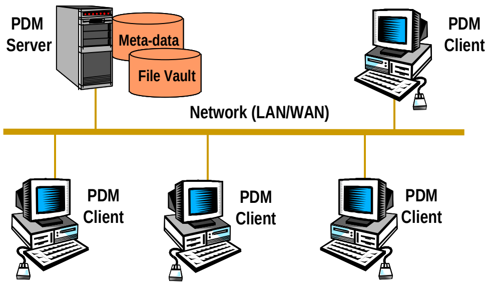


Figure 9 - Client/server principle

PDM systems run on different platforms, for instance, Windows NT and UNIX and some PDM systems can even run on mixed platforms, *i.e.* Windows NT and UNIX. Today there are also systems whose architecture is based on the Internet. The PDM server software is written in JAVA and runs on a JAVA virtual machine, which makes the PDM systems software platform independent. The use of Internet Browsers as a client makes the implementation rather simple; every computer with an Internet Browser can theoretically connect to the PDM system. All the required functionality resides at the server side. For the sake of performance additional software, so called plug-ins, is sometimes required on the client side.

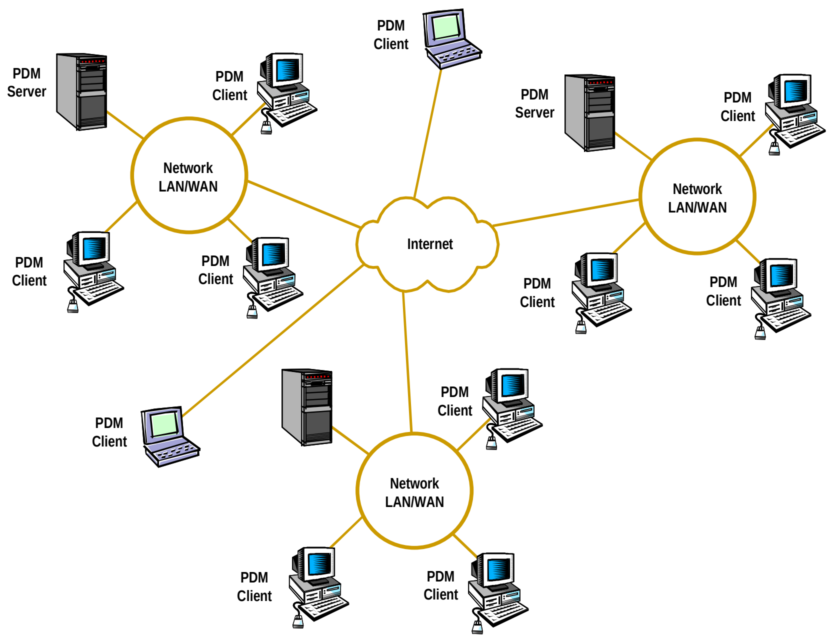


Figure 10 - Complex architecture of a PDM system

In practice this can result in a rather complex PDM architecture as is shown in Figure 10. It shows three PDM configurations, which are each composed of a PDM server and some clients that are connected via a network. These 3 PDM systems could represent the PDM systems of different workgroups of the same company and these systems do not necessarily use the same PDM software. These systems are connected via the Internet so that they can access each other's data. It is also possible to access data in the PDM with so-called remote clients (*i.e.* laptops) that dial-in to the Internet using an Internet Service Provider (ISP).

2.4 PDM related technologies

In this section several other technologies are briefly reviewed because they are related to Product Data Management systems. It is used to position PDM applications among all the other kinds of applications that can be used in product development organisations. Figure 11 presents an overview of the overlap of these different systems. The overlap is discussed in more detail in the following sections.

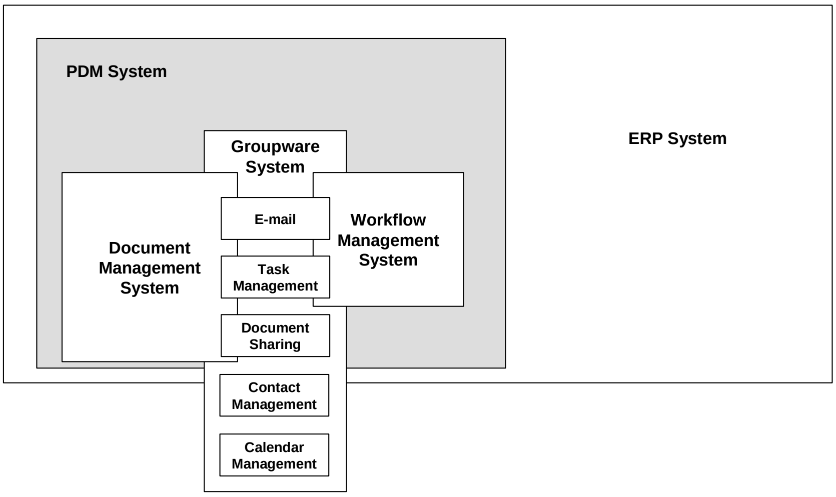


Figure 11 - Overlap between PDM and other technologies

2.4.1 Document Management

Document Management is one of the main functions of a PDM system (see section 2.2.2 and 2.2.4). However, the document management functionality that is offered by the major PDM vendors, such as SDRC, MatrixOne, PTC, and UGS, is different from the document management functionality offered by the major Document Management vendors, such as Documentum, Open Text and Filenet. The focus of the PDM vendors is on product development environments. Therefore, besides managing office documents they also offer possibilities to manage CAD-files and have integrations with CAD systems. These environments also require comprehensive version control functionality. This type of functionality is often not supported by Document Management systems because their focus is on the office environment.
It is not uncommon that large industrial organisations use PDM systems and Document Management systems together. PDM is then used in design and engineering for managing technical data while Document Management is used in the rest of the organisation to manage office documents.

2.4.2 Groupware

Examples of commonly used Groupware systems are Lotus Notes (Lotus), Groupwise (Novell), and Exchange (Microsoft). The functionality that is offered by these types of systems include e-mail, calendar management, contact management, task management, and document sharing.

Calendar and contact management are not supported by PDM systems. The other functions are to some extent supported by PDM systems. E-mail functionality, for example, is quite common in PDM systems. It is used to send messages to other users and these messages can contain links to objects in the PDM system. The first PDM systems had their own e-mail facilities. Today PDM systems can integrate with third party e-mail systems so those users have only one system for all their e-mail activities.
Task management involves the possibility to define tasks and the possibility to assign this task to one of the users. However, often this is not as advanced as the workflow functionality offered by PDM systems. Document sharing involves the possibility to store documents in so-called public folders. This means that employees who have access to this folder have access to the documents in this folder. This is not the same as document management in a PDM system where a document is put under control of the PDM system.
Because of the overlap between Groupware and PDM systems they can be used as complementary systems. Where the e-mail functionality of the Groupware system is completely integrated into the PDM system..

2.4.3 Workflow management

Examples of Workflow Management Systems (WFMS) are Staffware, Cosa and Visual Workflow. According to the Workflow Management Coalition (www.wfmc.org) the definition of workflow management is:

> *" The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."*

Van der Aalst and Van Hee (Hee, Aalst, 2001) defined that a WFMS handles the logistics of case handling. An example of a case in this context is a new insurance policy. The logistics then mean taking care of routing the insurance policy through the organisation.
Like ERP systems PDM systems contain workflow modules. A complete workflow management system can be part of a PDM system. Although the workflow management modules of PDM systems are often less sophisticated as stand-alone workflow management systems like Cosa. On the other hand a WFMS cannot be regarded as a PDM system because it does not offer other PDM functionality besides the workflow management functionality.

2.4.4 ERP

Examples of major ERP systems are MySAP (SAP), OneWorld (J.D. Edwards), Baan (Baan), and Peoplesoft (Peoplesoft). The predecessors of the ERP systems were the MRP systems that focused on manufacturing planning. Basically these systems determined the material requirements based on the Master Production Schedule and the Bill of Material of the products. Later other functionality was added to MRP systems; for instance functionality for financial control, accounting, and human resource management. This resulted in systems that are referred to as ERP systems today.
For a long time these systems hardly supported engineering or product development processes. However, during the mid 1990s the ERP vendors also discovered this market and offered PDM functionality as part of their ERP system. Initially they offered very basic PDM functionality that was not so sophisticated as the systems from CAD vendors or independent PDM vendors. Today there are vendors that offer competitive PDM products. SAP for example offers a Product Lifecycle Management (PLM) module as part of its ERP solution.

3 Product Development Processes

3.1 Introduction

The goal of this section is to define product development, put it in context, and describe typical characteristics of the product development process.

3.1.1 Product development context

Basically product development concerns the definition and creation of new products. It is one of the most basic characteristics of human beings; they make a wide range of tools and other artefacts to suit their own purposes (Cross, 1994). In traditional craft-based societies designing is not really separated from making; that is to say, there is usually no prior activity of drawing or modelling before the activity of making the artefact (Cross, 1994).
In modern, industrial societies the activities of designing and making artefacts are often separated. One of the reasons for this is that products have become more complex and hence more employees from several disciplines are needed to design and make the artefact or product. The separation of the activity of designing from the activity of making has also been stimulated by the Scientific Management principles, *i.e.* division of labour, from Frederick Taylor.

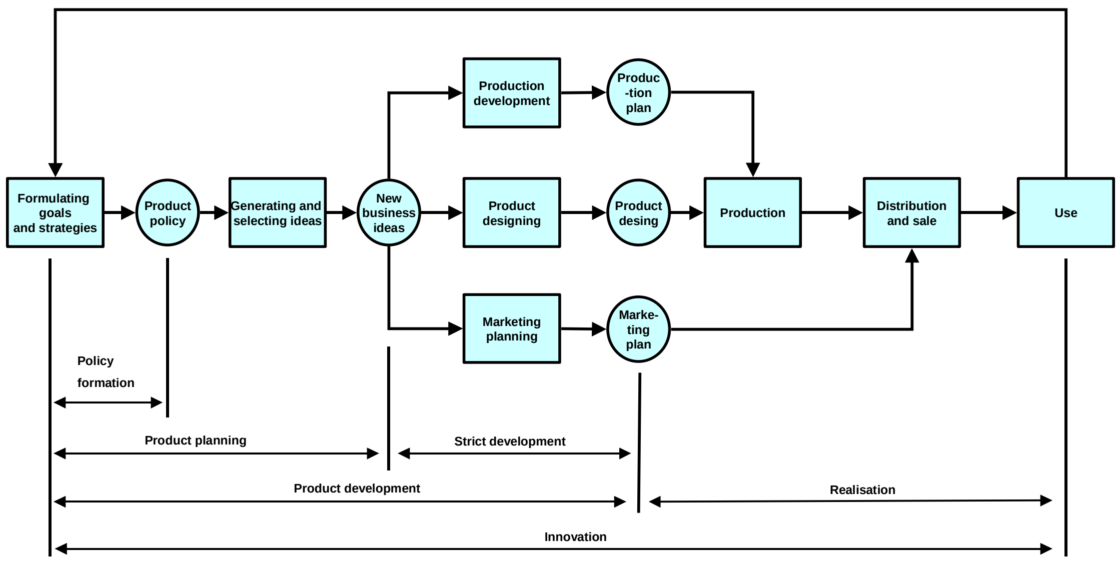


Figure 12 - Product Development in context

Product development is one of the processes of a company. Several authors such as Andreasen and Hein (1987), Hales (1991) and Roozenburg and Eekels (1991) provide models that put the product development process in context. De Graaf (1996) refers to these types of models as management models. The management models of different authors are quite similar. The model of Roozenburg and Eekels is selected to discuss product development in more detail because of its focus on processes (see Figure 12).

Roozenburg and Eekels use the name Innovation for the collection of all processes of a company. The Innovation process starts with what they call Product Planning (the first two processes). It involves defining the strategy and goals of a company and defining which markets they want to serve. Generating ideas and selecting ideas follows the definition of strategy and markets. The ideas concern new products and services or complete ranges of products and services. Wheelwright and Clark (1993) use a funnel to describe the generation and selection of ideas. Reinertsen (1997) calls it the Fuzzy Front End. With the term fuzzy he refers to the fact that during product planning it is often unclear what the market or customer needs are and how these needs can be translated into unambiguous product specifications.

Strict Development follows product Planning, there are two important processes in this phase that are conducted concurrently (see also Andreasen and Hein, 1987). It involves Technical Development (Production Development and Product Designing) and Commercial Development (Marketing Planning). Technical Development concerns the definition of form, fit, and function of the product as well as the definition of the required tooling and equipment to make the product. The process translates the selected idea from the Product Planning phase into a blueprint of the product. The Commercial Development process takes care of the business economic goals of the company. It determines how the company can make profit by developing and selling products and how the company can survive in the long run. The final process is the Realisation process. It concerns the production of the product, distribution and sales of the product, and the actual usage of the product by the customer. Although it is not shown in the figure it also includes maintaining the product.

3.1.2 Definition of Product Development

In literature several definitions can be found, which define the product development process in more detail. De Graaf (1996) reviewed definitions from Hales (1991), Finkelstein and Finkelstein (1993), the English Department of Scientific and Industrial Research (1964), and Wallace (1993) to come to a working definition for product development in his thesis:

> *"Product Development is a sequence of design processes that converts generally specified market needs or ideas into detailed information for satisfactory manufacturable products, through the application of scientific, technical and creative principles, acknowledging the requirements set by succeeding life cycle-processes[4]."*

This definition limits product development to Idea Finding and Strict Development according to the model of Roozenbrug and Eekels. It defines that product development does not include the manufacturing processes. The primary goal of a product development process is to provide all the required product-definition information, but also to ensure that the information is fit-for-use (Hamer and Lepoeter, 1996).

Although product development does not include manufacturing the definition mentions that it should acknowledge the requirements set by succeeding life cycle-processes such as manufacturing and service. Therefore, the result of product development is an intelligent trade-off between the various other design goals, for instance manufacturing cost, ease of use, and environmental aspects (Hamer and Lepoeter, 1996). Several techniques have been developed to support this way of thinking and is generally referred to as Design for X techniques (Tichem, 1997).

In this research the focus is on what Roozenburg and Eekels call Strict Development, because Product Data Management technology is hardly used during the earlier phases. During these phases other methods and tools are required. To define Strict Development the definition by

[4] *Life Cycle Processes* or the *Product Life Cycle* involves all the phases from conception through design, engineering, manufacturing, use and maintenance to disposal.

De Graaf is slightly adapted to exclude the Idea Finding phase, resulting in the following definition:

> *"A sequence of design and engineering processes that converts general product specifications into a detailed product definition for satisfactory manufacturable products, through the application of scientific, technical and creative principles, acknowledging the requirements set by succeeding life cycle-processes*."

This definition indicates that idea finding is not included. The activities, to be performed in the strict development phase, are design and engineering. Design determines the rough outline of the product while engineering determines the detailed product definition. The outcome of Strict Development has been defined here as *product definition* instead of the more general term *information*. The product definition involves all the information that is required to describe the product as well as the equipment and tooling that is required to make these products.
In the remainder of this thesis the term Product Development will be used instead of Strict Development because most readers are more familiar with the term product development.

3.1.3 What is product development and what is not?

The term product development suggests that the output is a new or innovative product. Booz, Allen and Hamilton (1998) defined newness according two dimensions: *new to the company* and *new to the market*. Viewed on a two-dimensional map, six different types of new products have been identified (see Figure 13). For a description of each type of product please refer to Booz, Allen and Hamilton (1982) or Cooper (1993).

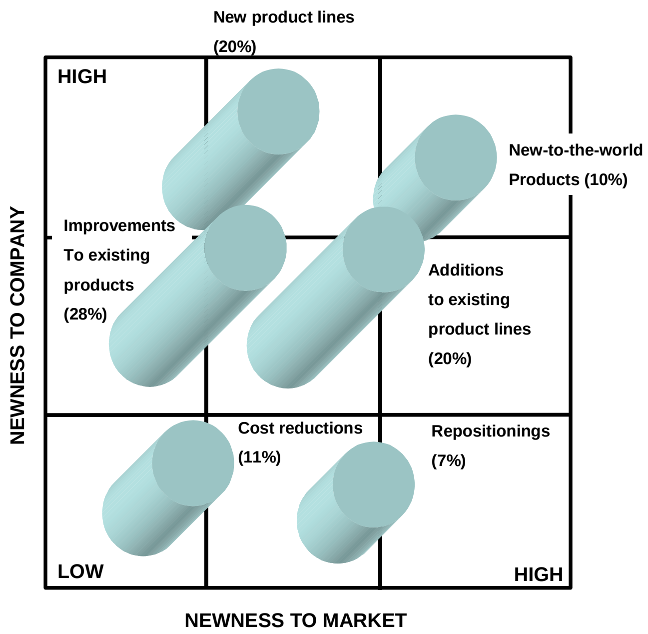


Figure 13 - Different types of 'new' products

The figure is used to indicate that product development not only results in innovative products. In figure A innovative products only involves *new-to-the-world* products. But the figure shows that product development can also result in less innovative products. For example, refining existing products is also product development, because the result of refining existing products is a new product definition that is used by production to produce the refined product.

Product development is also often confused with Research & Development. In this thesis Research & Development is defined as fundamental research that does not result in 'new' products. The output of Research & Development is a new technology, new functionality or a new material, which is input for Product Development to apply in new products. The duration of a R&D activity is typically longer than that of a product development activity. It can take more than 10 years to develop a new material or technology while it often takes ('only') several months to several years to develop and bring a new product to the market.

Product Development is often interpreted as the development of products that are produced in (large) series, for example cars, consumer products, and aeroplanes. However, there are also companies that produce One-of-a-Kind products, for example ships, satellites and production machines. Each product is unique because it is made to the specifications of the customer. Wortmann *et al.* (1997) refers to this phenomenon as '*Customer Driven Manufacturing*'. The major differences between product development for production in (large) series and product development for one-time production are:
- the input for product development is a market analysis respectively a customer order
- the product development activities focus on process design respectively product design

However, the outcome of product development in both situations is the product definition that is required by production to be able to produce the product. It is expected that both will handle starting on preliminary information in a similar fashion. Therefore I make no difference between these to types of product development processes in this research.

3.1.4 Typical characteristics of Product Development Processes

Product development processes (PDP) have typical characteristics. Only those characteristics that are relevant for this research are discussed in this section, it involves:
- Uncertainty (Wheelwright and Clark, 1993), (Muntslag, 1994), (Hoevenaars, 1994)
- Iterations (Roozenburg and Eekels, 1991), (Wheelwright and Clark, 1993), (Clausing, 1994)
- Product Complexity (Wheelwright and Clark, 1993), (Muntslag, 1994), (Korbijn, 1999)

Uncertainty

At the start of a product development project there can be a lot of uncertainty about the future. This is illustrated using an example of Wheelwright and Clark (1993):

> *"At a fundamental level the development process creates the future, and that future is often several years away. Consider, for example, the case of a new automobile. The very best companies in the world in 1990 could develop a new car in three to three and a half years. At the outset of a new car development program, therefore, designers, engineers, and marketers must conceive of a product that will attract customers three years in the future. But that product must also survive in the marketplace for at least another four to five years beyond that. Thus the challenge is to design and develop a product whose basic architecture will continue to be effective in the marketplace seven to eight years after it has been conceived."*

Therefore, at the start of a project assumptions are made about the customer requirements, about the demand for the new car etc. Uncertainty comes to the surface when there are changes or unforeseen events. Examples of changes or unforeseen events are changing customer requirements, lower demand or new technology or materials that becomes available. When changes or events occur a company has to revise its assumptions and has to determine the new course of action.

This can lead to iterations in the product development process. For instance, if new materials and technology become available during the project then a company has to make the decision whether to apply this new material or technology or not[5]. If they decide to apply it they have to re-do part of their product development process what delays the introduction of the product to the market.

Iterations

In the previous paragraph it was already identified that uncertainty can lead to iterations in the product development process. Clausing (1994) identified two types of iteration that are directly related to the design work itself, *creative* iteration and *dysfunctional* iteration.

Creative iteration is the result of the fact that it is impossible to conceive an idea, consider it from all perspectives instantaneously, and draw the final design. Several iterations are required to consider an idea from several perspectives, which can lead to several alternatives. During the iterations the idea slowly evolves to the final design, *i.e.* select one of the alternatives.

Dysfunctional iteration involves amongst others the stream of changes caused by problems regarding the manufacturability and maintainability of the product or by additional customer requirements. Consequently designers and engineers have to revise their work and that takes time. The later a problem is detected in the project, the longer the iteration and hence the delay. Hence in contrast to creative iterations, dysfunctional iterations do not add value but incur unnecessary cost.

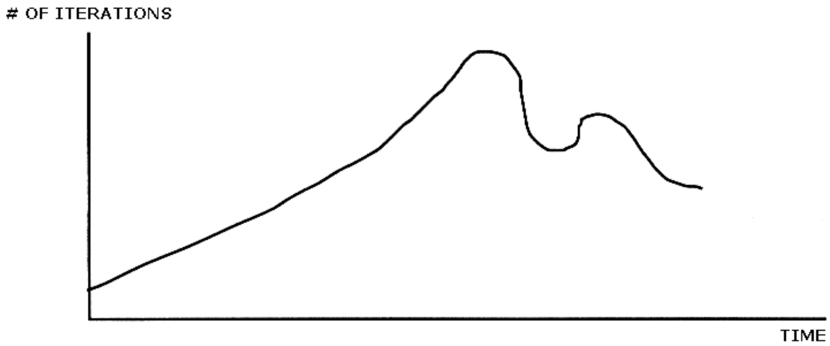


Figure 14 - Bad design of a product development process (Simonse, 1998)

A badly designed product development process can be recognised by looking at the number of iterations over time. Figure 14 shows an example of such a bad designed product development process. The number of changes increases during the development process and reaches its highest point during production. In production all errors made during design are found. Then the number of changes drops because the product needs to be shipped to the market/customer. After that the number of changes increases again because during use of the product new errors are found that were not discovered during design and production. All the changes lead to expensive iterations and hence delay the project.

[5] Clausing (1994) refer to this phenomenon as the technology stream. Because of the opportunities companies should constantly monitor this technology stream.

Product Complexity

Over the last decades the complexity of products increased, amongst others because of the application of electrical and software components. The increased product complexity also affects the number of people that are required to develop a product. Take for instance an aeroplane; Sabbagh (1995) described a Boeing 777 as '*several million of parts flying in close formation*'. Because of the amount of parts it took thousands of employees to develop a Boeing 777.

An example from Korbijn (1999) also illustrates product complexity. He compared a copier from the mid nineties with a copier from the mid eighties (Table 2). Both copiers are able to copy 45 black/white A4 pages per minute. However, the engineers have added much more functionality to the copier from the mid nineties. The following table shows a comparison of the number of components in both copiers as a measure for the complexity. Because of the increase in parts also the number of employees increased substantially, from 500 to 1250.

Table 2 - Complexity of a copier

	mid 80's	mid 90's
Number of different mechanical parts	3.300	4.200
Number of electro-mechanical parts	19	77
Number of sensors	15	52
Number of components on the main control board	443	614
Number of programmable components	1	7
Source code in Mb's	4.5	12.5

The complexity increases more than exponentially as the number of parts increases because the number of possible interfaces between the parts also increases exponentially. Therefore good interface definition is an important activity in product development because it can reduce the number of interfaces (Reinertsen, 1997).

But not only the number of parts determines the complexity of a product. Also the shape of a product or the number of constraints can increase the complexity of a product. Take for instance a razor, which consists of a small number of parts but is very complex to design.

Note: In electronics there is a trend towards more functions on a single component (Very Large Scale Integration, VLSI). Hence the number of components decreases, but the product complexity increases because the number of functions increases. Hence in electronics the number of functions is a better measure.

3.2 Product definition information in product development

This section describes the role of product definition information in product development. The focus is on the documents that flow through the 'Design Factory' and especially on the release process of these documents.

3.2.1 Introduction

The product development process creates product definition information that describes the product at several stages in its evolution. Examples of this information are specifications, drawings, digital mock-ups, stress analysis reports, NC files, parts lists, drawing lists, and change requests[6]. The amount of product definition information that is required to describe a single product has increased during the last decades. There are several factors that led to the increase of product definition information. First of all because of the increased product complexity (*i.e.* increasing number of parts and components) there is more concerning a product that needs to be documented (Hameri, 1998).
Other factors include the increasing amount of standards to which a product should comply, product liability laws (specifying exactly what the product will and will not do), the increasing awareness of quality issues (producing quality user manuals), user documentation, and maintenance documentation (McIntosh, 1995). Finally, the introduction of computer applications led to an increase in the volume of product definition data (Sackett *et al.*, 1998). Because of these computer applications it became easier to create product definition information.

This increase in product definition information is being experienced in all manufacturing and service sectors but it is especially apparent in defence and defence related industries (McIntosh, 1995). To illustrate this McIntosh (1995) mentions a US Navy warship example. The associated design and service-related documentation of the warship actually consumed more space than the warship itself. Aerospace industries have also quoted figures greater than 1:1 in relation to the weight of associated documentation versus the weight of the product.
Today product definition information is no longer solely produced on paper, microfilms, or polyester. Because of the introduction of computer-based applications, such as CAD and CAE, product definition information has also become available in all kinds of electronic formats. The introduction of computer-based applications introduced a larger variety of formats that need to be managed, besides paper documents, microfilms, and polyester's, other different electronic formats need to be managed today.

A product's information needs to be available throughout its entire (technical) lifecycle, because it can be in the field for over 25 years. After 15 years for instance spare parts might be needed and hence the drawings to make these spare parts. In case there is an accident with the product after 20 years the product information is required to show that it complies with all applicable regulations. Concluding product definition information is an important asset that needs to be managed over the complete lifecycle of a product. Because of the increasing amount of information and the increasing amount of formats managing this information becomes more critical.

[6] For a more complete list of product information that typically is generated during the lifecycle of a product please refer to Vroom (1996).

3.2.2 The Design Factory

Reinertsen (1997) compared the product development process with a factory resulting in the term *Design Factory*. Instead of producing products the Design Factory produces product definition information, *i.e.* designs, that describes the product.

The product definition information in the product development process is mainly created in the form of documents, either analogue or digital. Specialists in the product development process create documents. To create a document specialists need input from other engineers in the form of documents. This exchange of information between actors in the product development process results in a stream of documents that flows through the product development process (see Figure 15).

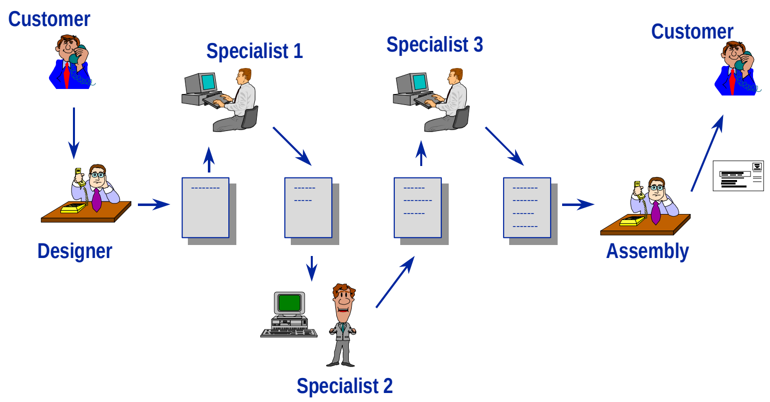


Figure 15 - The Design Factory

Before a document can be released for wider use it needs to be checked and approved by someone else (Peltonen *et al.*, 1996). This process ensures the quality of the document that is used as input by other activities. A low quality of the released document would result in iteration loops that are required to correct the low quality, *i.e.* error. The document release process is an important part of the product development process that is described in more detail in the following sections. But first the term document version is introduced and explained in more detail because it is crucial for explaining the document release process in more detail.

3.2.3 Document Version

In this section document versions are discussed in more detail by looking at several aspects of document versions. It starts with introducing two definitions of the term (document) version.

3.2.3.1 Definition of document version

Peltonen *et al.* (1996) mentioned that versioning is one of the most important, and ambiguous, concepts in engineering data management (*i.e.* product data management). In their paper they use the definition of Katz (1990):

> *"A version is a semantically meaningful snapshot of a design object at a point in time. As such it is a descendent of some existing versions (if not the first version) and can serve as an ancestor of additional versions. Although every change could result in a new version, this is usually not the case. New versions are created as part of the design process: Changes are made, the changed object is verified, and if these changes are deemed acceptable, a new version is created or released."*

Hamer and Lepoeter (1996) defined versions as follows:

> *"Designers typically modify design information in multiple steps. Each steps results in a new version of the design information. Sometimes a designer needs these steps to add functionality in a controlled way, e.g. when creating a complicated design from scratch. In other cases, versions correspond to the steps that designers take in correcting mistakes, or in optimising the design. However, in all cases a new version is created because the designer wants to modify the design."*

Both authors have a similar definition of versions. However, Katz has a more formal description of versions while Hamer and Lepoeter have a more pragmatic definition because they directly relate it to the design process. What is important is that both authors mention that a version of a design object is created by deriving it from the original. The content of the original (or previous version) is used as the starting point for the new version of the object. Therefore, there are relationships between versions of an object resulting in a version tree. Such a version tree shows the history of an object (chronological order). This explains why Katz mentions that a version should be a meaningful snapshot at a point in time. It is not useful to create a new version for every small change. It is only worthwhile to create a new version if the change should be recorded so that it appears in the history of an object.
The version mechanism that is described for objects can also be applied to documents. Applying the mechanism to a document means that there can be several versions of a document. By introducing document versions the term document becomes ambiguous (Peltonen *et al.*, 1996). Therefore I distinguish between an *abstract document* and its *document versions*. At one point in time an abstract document can be represented by different document versions. This depends on the person that looks at the (abstract) document. Therefore, there is no such thing as the last document version because it depends on the person/user that is looking. This has to do with the effectivity of a document version. An engineer for instance would see the *highest* document version while a manufacturing engineer would see the *latest released* document version (discussed in more detail in section 3.2.4.5).

Summarising the following definition of a document version is used in this thesis:

> *"A document version is a meaningful snapshot of the content of an abstract document at some moment in time. Document versions of the same abstract document show the history of an abstract document over time (in chronological order)."*

3.2.3.2 Use of document versions

In the previous section it was mentioned that it is only worthwhile to create a new version if the change should be recorded so that it appears in the history of the document. In our opinion there are at least 3 reasons that make it worthwhile to create a document version:

- *Point of reference (or baseline)*: Hamer and Lepoeter (1996) mention that designers modify a design in multiple steps and that they need these steps to add functionality in a controlled way, *e.g.* when creating a complicated design from scratch. At the end of each step they want input from others about the design before they continue with the next step. Therefore, they create a version of the design document that they send to colleagues for comments. If comments come back to the designer they know to which version of the document the comments apply (*i.e.* you have a common point of reference). If you make the changes as suggested by the reviewers you create a new version and send that once again to the reviewers. Now the reviewers can see that they received a successive version of a document that they have reviewed before. By comparing the versions they can see what has been changed or added to the design.

In this case the version is a snapshot of the design document at an important point that serves as a point of reference or baseline for everybody that is involved.

- *Ability to undo changes*: Imagine that you are working on a report. In the beginning the document is empty and after 8 hours of work there is a first section. When you continue the next day you might decide to remove a paragraph from the first section after which you continue with the second section. The third day you might decide to include the paragraph that you removed in chapter 3. If you continued working on the same document after deleting the paragraph, the paragraph is no longer there and you have to retype the paragraph. However, if you decided to make a new version before you deleted the paragraph you have the possibility to go back to the previous version of the document and to copy the paragraph to the current version.

- *Audit trail to track evolution*: There are regulations such as for product liability that demands that a company can show what happened to a particular design during its lifecycle. In case there are any accidents with a product, the companies that developed and produced the product have to proof that they are not to blame. If companies did not store the version history of their design documents they have a hard time proving their 'innocence'. If they did store the version history of the design documents they are able to go back in time and see why certain decisions were made.
  In this case versions are created at moments in time when important decisions are taken with respect to the design document. Hence it serves as an audit trail to track the evolution of the document. In section 3.2.4 is mentioned that document versions are created at every change of the document status. The reason is traceability of the document through its document lifecycle.

3.2.3.3 Document version numbering

Document version numbers are used to distinguish between several document versions. As document versions are snapshots of the same document they have the same document number which makes it hard to distinguish between them. The combination of document number and version number are used as a unique identifier for document versions.
In practice there are several mechanisms for version numbering. Some companies use 1, 2, 3, 4 … and others use A, B, C, D … for numbering successive versions. Figure 16 shows a simple example of successive version numbering using a so-called version tree.

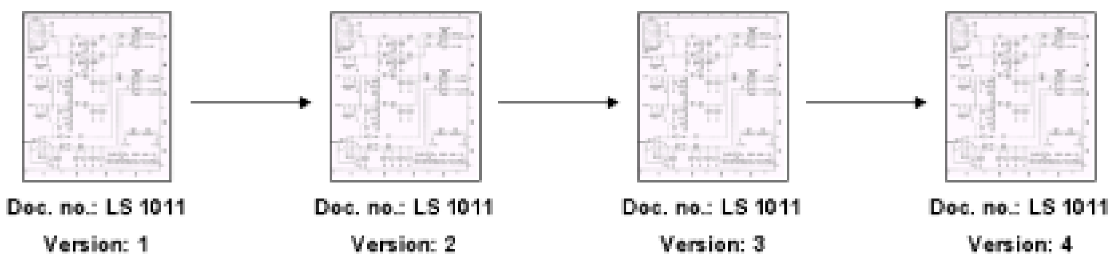


Figure 16 - Successive document versions

It shows the successive versions of a drawing with document number LS 1011 and it shows that that version 3 is the successor of version 2 who is the successor of version 1. In other words the arrows represent the derivation relationships between the different versions.

Peltonen *et al.* also shows that it is possible to have branches in the version tree, for instance if parallel versions are required (splitting of branches).

Sometimes companies use more than one version number in order to distinguish between internal and external versions. Internal versions are used to store intermediate results of internal iterations and external versions are used to communicate with the customer. Then 1, 2, 3, 4 is for instance used for internal versions and A, B, C, D for external versions. The internal versions then go from 0.1, 0.2, 0.3. When the document is send for review to the customer it receives version A, not 0.3 because the customer is not interested in the internal revisions of the document. If the customer has any remarks the company has to adapt the document and the internal versions then go from A.1, A.2. If the document is once again sent to the customer it will receive version B. This example shows that the version number of the document depends on the viewpoint of the person that looks at the design. An employee of the company would see A.2 while the customer only sees B.
This mechanism can easily be extended if a company wants to distinguish between more than two types of versions. If a company wants to distinguish between three types of revisions the version number on the lowest level might look like A.1.3. Summarising multiple levels in the version number represent multiple levels of iteration in the product development process.

3.2.3.4 New document or new document version?

When a new abstract document is created, a new, unique number is created to identify this abstract document. If changes are made to this document it results in new version of the abstract document that have the same identification number but different version numbers. According the definition presented in section 3.2.3.1, document versioning is used to create meaningful snapshots of a document over a period of time. Hence there is a successive relationship between versions of the same document, where the old version evolved into the new version. A typical characteristic of versions is therefore that they are interchangeable, the old version can be replaced by the new version. If a change to a document version does not result in a successor of that document version, one should create a new document instead of a new document version. This new document can not be used instead of the other document; *i.e.* they are not interchangeable.

In manufacturing they use the Form, Fit, Function (FFF) rule to tackle a similar problem concerning part numbering. When the Form (*e.g.* shape), Fit (*e.g.* tolerances) *or* Function of a part is changed a new part number is created. Otherwise a new version of the part is created and the parts are considered to be successors of each other. In that case the old version of the part can be replaced by the new version of the part without any consequences.
But why is this so important in manufacturing? Each part has to be supported during the entire product lifecycle, in case the part damages it should be possible to replace the part. Supporting a part during the entire product lifecycle is expensive for several reasons, amongst others administration and the fact that you have keep each part in stock. If a new version of a part is created it is not required to keep a stock of the old part anymore because the new version can replace the old version if necessary. On the other hand, two parts with different part numbers have a different Form, Fit and Function and can therefore not replace each other. Hence it is required to support both parts during the entire product lifecycle, which is more expensive.

For a number of documents there are direct links to parts, because they are used to capture the Form, Fit or Function of a part or a combination of those aspects. In that case it is possible to apply the Form, Fit, Function rule to documents by coupling the version numbering of parts and documents. For example, a drawing is used to capture the geometric information (*i.e.* Form) of a part. When the Form of the part is changed it is required to change the drawing, because it captures the Form. Therefore both the part number and the drawing number should change. On the other hand when the Function of the part changes but not its Form, the part number changes but not the drawing number.

However, there are also a lot of documents that are not directly related to a product. In that case it is important to assess whether a change leads to a successive document version or into a complete document. This is illustrated using an example. A change in a memo X that is used to introduce employee A to the company results in a new version, if the memo is still used to introduce employee A. In case memo A is changed at a later point in time to introduce new employee B one should create a new document because the memo is not used anymore to introduce employee A. In other words, there is no successive relationship between the memos and they are not interchangeable. Obviously one should re-use memo A as much as possible, but that does not change the fact that one should create a new document (number)[7].
The same rules can be applied to the question whether the change in the layout or language of a document should result in a new document version or a new document. Such a change does not affect the content of the document but only the layout. Hence they can be considered as successors that are interchangeable, *e.g.* a quality manual with a new layout replaces the previous manual. But what should be done if it concerns a drawing that needs a different template for two different customers? In this case one should create two different drawings, with different drawing numbers. The reason is that both drawings are not successors of each other. Moreover, they are not interchangeable because customer Y will not accept the drawing for customer X because the template is wrong.

Finally, I would like to address one more issues on this subject. It concerns 'variants' and 'alternatives' that are easily confused with versions. Version numbering is often abused to identify different variants or alternatives. In my opinion this is not correct for the following reasons.
Variant is a term that is used in product family modelling. Within a product family there can be different variants of a product that are based on the same product architecture (Erens, 1996). Examples of variants in manufacturing are cars that can be ordered with or without air conditioning, in different colours, or in a sports edition or a executive edition. Translated to the document world, one could think of a service manual that is available in different languages. In my opinion they should have different document numbers because they are not interchangeable (a condition for version numbering). Somebody in Japan can probably not use a manual that is written in the Dutch language, because he does not understand Dutch. In a PDM system one should use a '*is a variant of*' relationship to indicate that two documents are two variants instead of two versions.

---

[7] An exception here is when the content and the format of the document are separated and can be generated on demand. This is the case when you use the Mail Merge function of Word. The new employee name is filled in as a variable and Word generates a letter (from a template) and fills in the name of the employee in some predefined places, i.e. fields, in the document. In this case the document number of the template is not changed. To generate the letter it is only required to store which letter format was used to generate the introduction letter for a specific employee.
N.B. In this context one could also think of XML applications that are used to separate content and format.

Alternatives are typically used during the design phase of a product and represent several alternatives for the same product. Alternatives exist next to each other and one of the alternatives is selected after a (careful) selection process. Therefore it is not justified to call alternatives successors of each other or interchangeable. Therefore one should different document numbers to identify the alternatives instead of version numbers. In a PDM system one should use a '*is an alternative of*' relationship to indicate that two documents are alternatives instead of a version number.
Summarising, if a document captures the Form, Fit or Function of a part or a combination of those aspects, then the Form, Fit, Function rule can be applied. If the Form, Fit or Function, that is captured by the document, changes then one should create a new document, if not one should create a new document version. If a document does not capture Form, Fit or Function aspects of a part then one should assess whether a change leads to a successive document version or into a complete new document. In the case of two document versions they are interchangeable.
Finally, document version numbering should not be used when it involves variants or alternatives. In that case different document numbers should be used to identify the different variants or alternatives instead of document versions.

Note: The intention of this section is not to be complete and I can imagine that also other considerations play a role to determine whether something is a new document or a new document version. By means of this section I only intend to provide my point of view to start a discussion. Anyhow, for my thesis it is not important to have rock-solid definition. The definition that a document version is meaningful snapshot of the content of a document at one point in time, is sufficient.

3.2.4 Document release process

3.2.4.1 Introduction

Now document versions have been introduced it is time to return to the document release process. First the document release process will be discussed at the abstract document level, to explain the purpose of the release process. After that the release process is discussed in more detail which requires the introduction of the term document version.

At the end of section 3.2.2 it was mentioned that documents needs to be checked and approved by someone else before they can be released for wider use (Peltonen *et al.*, 1996). Hamer and Lepoeter (1996) come to a similar conclusion:

> "*Organisational procedures are used to maximise the likelihood that the design is satisfactory. Familiar examples of such procedures are the introduction of verification steps (e.g. electrical simulation) and validation steps (e.g. field-testing) and approval procedures (releasing). When design information passes such a step it results in a change in status … These procedures ensure that the document goes from being tentative, untested and possibly incorrect to stable, consolidated, and proven.*"

These procedures describe the document release process in the product development process. Such procedures are typically used to increase the quality of the documents before they are released for wider use. Several authors have described the release process of documents; examples are Bos *et al.* (1987) and Harris *et al.* (1997). Quality standards such as ISO 9001 also prescribe document release procedures. A release process consists of at least two activities of the following type:

1. review/check
2. authorisation and freeze

The release process starts when the creator of the document considers it to be complete and submits it for review. The purpose of the review step is to check (Bos *et al.*, 1987):

- the correctness of the content
- the basic assumptions that are used
- the acceptability of the content
- the consistency of the content (interfaces)
- the applications of standards

A review detects a number of 'errors' in a document before it is released for wider use. Hence a review step increases the quality of the document. In a review employees from several disciplines check the document. This can result in several review steps that are executed sequentially. Moreover it is not uncommon that the customer is also involved in the review step.

After a successful review step(s) the document is considered to be stable, consolidated, and proven and released for wider use. From this moment on changes become increasingly more expensive, therefore the content of the document is often frozen. Released information is a stable input for other activities; as long as documents are not released they are prone to changes and errors.

Due to the uncertainty in the product development process it might be required to change documents once they are released. Companies use change control procedures to manage these changes. Once a document is released (or frozen), a document can only be changed by means of a change control procedure. This means that an engineering change request has to be submitted to a committee that evaluates the request. If they agree that the change is worth all the effort (*i.e.* cost and delay) the change is made to the document. Document freeze is an important aspect that will be addressed again when discussing starting on preliminary information.

3.2.4.2 Document Lifecycle

In this research all the steps that precede the final release of the documents are considered to be part of the document release process, these steps together form the *document lifecycle*. The quality of a document increases as a result of each step or as Hamer and Lepoeter (1996) state:

"the document goes from being tentative, untested and possibly incorrect to stable, consolidated, and proven".

This increase in quality is captured in a so-called *document status*. Different versions of the same document may have different statuses. Therefore it is incorrect to speak of the status of a document: a document with different versions does not have a unique statuses, only the versions have. The status of a document version says something about its quality and what can or should be done with that document version.

During its release or lifecycle a document version is promoted from one status to the next. All possible statuses of a document version in the document lifecycle can be modelled using a *state transition diagram* or *state graph*. Such a state graph is a graphical representation of the document lifecycle. Peltonen *et al.* (1996) use the example as shown in Figure 17. In this figure the circles represent the possible states or statuses in the document lifecycle. The arrows represent the possible state transitions in the document lifecycle, *i.e.* the promotion to a higher status. However this must be interpreted as the transitions of the latest version of the document.

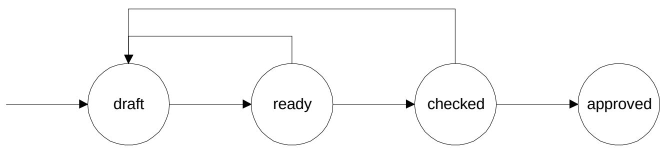


Figure 17 - Document lifecycle according Peltonen

Figure 17 shows that a document is first in the *draft* state. This is the only state in which the document can be modified. After the person who created the document considers the document complete, he moves it into state *ready* and submits it for checking. If the document passes the check it is moved to state *checked*. The final step in this document lifecycle is the document approval or release step in which the document is authorised, *e.g.* by putting a signature on it. The result is that the document is moved to state *approved*. Figure 17 also shows loops, for instance from status *checked* to status *draft*. If a document is rejected during the check it does not receive the status *checked*, but returns to the status *draft*. In other words the author has to modify the document to correct the errors that were found during the check.

Summarising the statuses in this example have the following meaning:
- *Draft*: document is edited by the author
- *Ready*: the author finished working on the document
- *Reviewed*: the document was successfully reviewed, *i.e.* not rejected
- *Approved*: the document is successfully approved and hence released for wider use

This state graph is a good example to illustrate that two things are often mixed up: *process status* and *document status*. The *draft* state for instance, sometimes called *in work* or *in progress*, is a process status while the others are document statuses. Peltonen *et al.* defines the *draft* state as the state in which the document is being modified. The other states, however, do not describe the process but the outcome of the process; *i.e.* the document is *checked* or *approved*. In my opinion the document status should represent the outcome of a certain process or step and not the process itself. A document should receive a status when it has reached a (intermediate) stable state.
When a document version is being processed it is not available to others, only when it has reached the next steady state does it become available to others. *Ready*, *checked* and *approved* are considered to be steady states here while *draft* is an unsteady state because the document is being processed and therefore, changes constantly. However, this does not mean that process statuses are not important. On the contrary, in fact both document status and process status should be monitored. Examples of process statuses are *in work*, *waiting for capacity* or *waiting for information*. The process status is an indication of the progress of the document with respect to reaching the next status in the document lifecycle. While the document status gives an indication of the maturity or quality of a document.

In a PDM system processes are often modelled as a workflow, while the document status is modelled as an attribute of a document. There is a connection between workflow and document status because a workflow can result in a change of document status and a change in document status can trigger a new workflow. If the document release process is modelled in a PDM system a user can look up the document status, for example *ready*. If it is *ready* it means that it is submitted to the review process (see Figure 17). By consulting the review workflow of the document he can see if there is any progress in the review process. An example of progress would be that 2 out of 3 reviewers already gave their comments. This information should not be used as a document status because it tells the user something about the process. In the early days when PDM systems did not (yet) support workflow the process status was also modelled as a document status. This is probably the reason why these two are often mixed up.

To close this discussion on what is a document status and what is not one last issue is addressed. It involves the check-in and check-out mechanism of documents. Check-out means that a document is checked-out of the PDM database. This means that the last version resides in the workspace of the author and is only accessible for the author. The previous version may remain accessible for others, but will be flagged that a new version is in process. Checked-in means that the author has finished working on this version and has put it under control of the PDM system. The latest version is accessible by all authorised persons. This mechanism prevents two people making a simultaneously change to the same document.
Check-in and checkout are sometimes used as a document status in the release process. However, in this research check-in and check-out are not used as document statuses, because they do not say anything about the quality of the document. They only tell something about the status of the document in the PDM database.

3.2.4.3 Document version lifecycle

Figure 17 shows loops, for instance from status *checked* to status *draft*. It suggests that a document could be promoted from a status with high quality to a status with low quality. In my opinion this is confusing and should be avoided. When a document can go back in status several times it becomes unclear for the user what has happened to the document. To make it possible to monitor the history of the document it is required to introduce document versions. A document version is always promoted to a higher status and never returns to a lower status. If it is required to return to a lower status a new document version should be created. In that case the user has access to the complete history of a document.
The following document version lifecycle model illustrates that a document version can not return to a lower status (Figure 18). It involves a so-called state-transition diagram that shows all the possible state-transitions of a single document version.

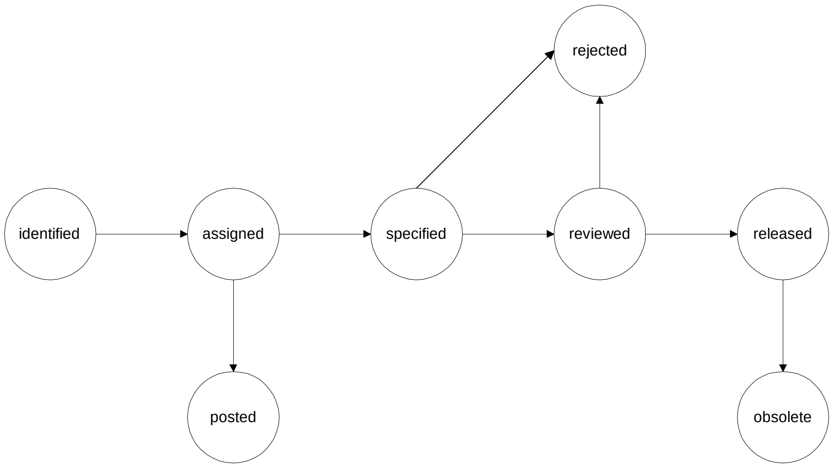


Figure 18 – State-transition diagram of a document version

The figure shows that the life of a document version starts with the *Identified* status and a document version can only be promoted to a higher status (see direction of arrows, *i.e.* state transitions). Finally, the figure shows there are 3 possible end statuses: *Posted*, *Rejected*, and *Obsolete*. If a document version ends in status *Posted* or *Rejected* it means that there is a consecutive document version. If a document version ends in status Obsolete it means that the document version may no longer be used for any purpose.

The document statuses as presented in Figure 18 are briefly explained below:

Identified: Document is identified and its document number is generated.

Assigned: Document version is assigned to an engineer who will create the content of the document.

Posted: Document version is posted, other engineers receive this version to provide feedback, this is a design activity not a check/review activity!

Specified: Document version is specified, according to the author the document is mature enough for the check/review process.

Reviewed: Document version is reviewed, there are no errors in the document and it can be submitted for release.

Released: Document version is authorised, an authorised person looked at the document and the comments of the reviewers and he agreed with the opinion of the reviewers and therefore, officially released the document.

Rejected: Document version is rejected, during the review or approval step the document has been rejected. The document including the comments is send back to the author who incorporates the comments in a new version.

Obsolete: Document version became obsolete; hence it is not allowed to use that document version anymore.

3.2.4.4 Document workflow model

Figure 17 in section 3.2.4.2 has loops and it was mentioned in the previous section that there should be no loops in a lifecycle model. The problem with figure 5 is that it tries to combine a model of the document workflow with a model of the document lifecycle. In this section is shown how a workflow model of an abstract document and a document (version) lifecycle model should be combined.

This is illustrated using the (abstract) document workflow model that is shown in Figure 19. In this model the boxes represent the (workflow) activities, the triangles represent waiting queues and the arrows represent the transition from a waiting queue to an activity and vice versa. Finally the vertical lines after an activity represent the document status (in this example the statuses as presented in the previous section are used). It indicates that a document's status changes as a result of an activity that acts on a document. The figure also shows two feedback loops, each feedback loop should result in the creation of a new document version.

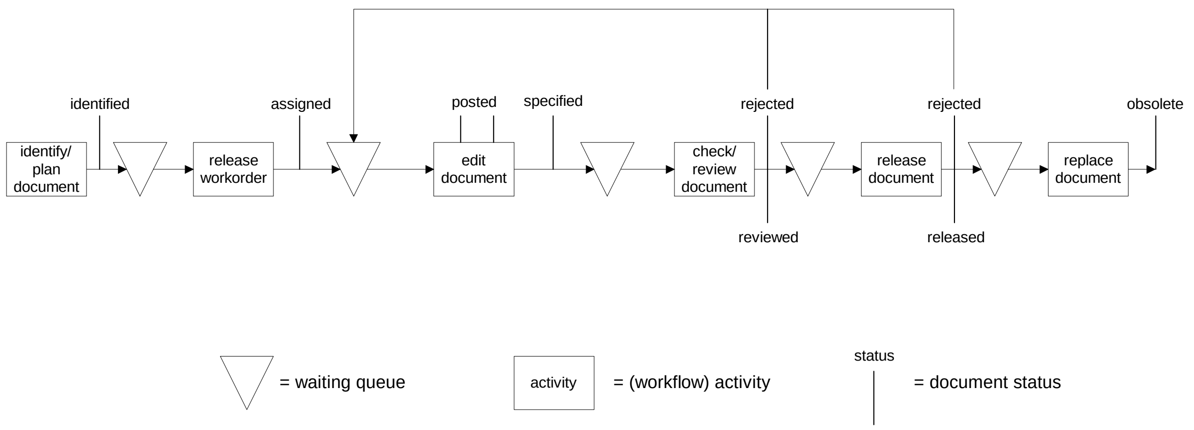


Figure 19 - Document workflow model

The document workflow model is discussed in more detail using an example of a manufacturer of packaging equipment. During a project, in which they have to develop and build a piece of packaging equipment, they identify the need to make a mono drawing (the start of the document lifecycle). An abstract mono drawing is created that receives the status *identified.* This means that it receives a drawing number. At this moment the drawing waits in the queue till it gets assigned to an engineer. When the drawing is assigned to an engineer it is once again placed in a waiting queue. If the engineer is available he picks one of the jobs in the waiting queue, *i.e.* his task list, and starts working on it. This means he creates an empty document with version number 1 and status assigned.

During the design process the designer also wants to have input from others before he finishes the document, *i.e.* he wants input before it is submitted for formal review. Therefore, he can post the document (several times) to a limited group of employees in order to get feedback. This enables the designer to do fast iterations resulting in a better design. Posting a document version for feedback means checking-in the version with status *posted*. The designer may immediately or later check-out a new version and continue working on it.

When the engineer finishes his work the mono drawing receives the status *specified* and is submitted for review, this means that it is placed in a waiting queue. The document stays in the waiting queue until the reviewer picks it up. This is the start of the review process where the reviewer checks whether the document is complete, consistent, and fit for use.
The review process can result in two decisions, *rejected* or *reviewed*. If the reviewer does not agree the document receives the status *rejected*. In this case the engineer has to make changes to (a new version of) the mono drawing according to the comments of the reviewer and has to re-submit that version for review. If the reviewer agrees to the mono drawing it receives the status *reviewed* and is submitted for release, this means that it is placed in a waiting queue. The document stays in the waiting queue until the authorised person picks it up.
This authorised person can release a version of the mono drawing to manufacturing engineering or production for example. If the version of the drawing is not approved it receives status *rejected* and is send back to the engineer. In this case the engineer has to make changes to a new version of the mono drawing according to the comments of the reviewer. After making the changes he re-submits the new document version for formal approval. If the new version of the drawing is approved it receives status *released*.
After a while the released drawing version might be replaced by a version of a completely new drawing (*i.e.* with a new drawing number and not only a new version of the drawing). The (latest) drawing version will then receive status obsolete, which means that it is replaced and may not be used anymore.

Note: The details of the review and approval process can vary per company; the above is just an example to describe the basic mechanisms.

3.2.4.5 Document history: status - version matrix

The document version lifecycle model and the abstract document workflow model do not provide a clear picture of the complete version history of the abstract document. This can be solved using the so-called *status-version matrix* that I developed. The status-version matrix is illustrated by modelling the document history of abstract document LS1101. The document versions in this example have a document version lifecycle that is shown in Figure 18.

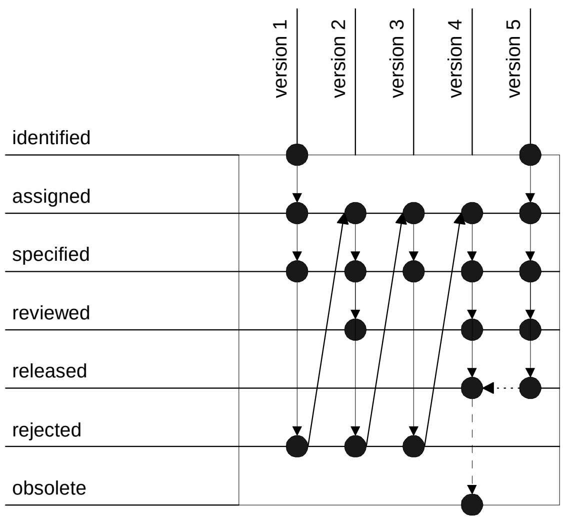


Figure 20 - Document status - version matrix

The history of abstract document LS1101 is shown in the status - version matrix in Figure 20. It shows that version 1 was identified, assigned and specified but that it did not pass the check/review and hence received status *rejected*. This resulted in the creation of version 2 that received status *assigned* and status *specified* as well. Version 2 passed the check/review and therefore received status *reviewed* but it did not pass the release step and therefore received status *rejected*. This resulted in the creation of version 3 that received status *assigned* and *specified*. This version did not pass the check/review and therefore received status *rejected* and finally version 4 was created. This version passed the check/review and release step and was therefore *released*. Then for some reason it is identified that version 4 needs an update, which results in the creation of version 5. However, until version 5 is released, version 4 will also remain active. The example illustrates that there can be more than one active version of an abstract document. Once version 5 is released it replaces version 4; at that moment version 4 receives the status *obsolete* (indicated with the dashed arrows in Figure 20).

The matrix can also be used to show the effectivity rule that was already mentioned in section 3.2.3.1 and 3.2.3.3. Not everybody in an organisation will have access to the latest version/status of a document. For example, when the document has status *assigned* it is only accessible to the author of the document. When it has status *specified* it is accessible to the reviewers because they have to check the document. When it is *reviewed* during the review process the document becomes available to the members of the discipline of the author. When it is finally *released* all other employees in the company also have access to the document.
The effectivity rule means that everybody in the company has access to the released document version 4 and 5 while colleagues of the author will also have access to the reviewed document version 2 and 4. This illustrates that an abstract document can be represented by several document versions at the same time. The document version that represents the abstract document depends on the user that is looking at the abstract document.

3.3 Concurrent Engineering

3.3.1 Introduction

Concurrent Engineering emerged during the eighties of the previous century. By that time the U.S. and Western European manufacturers found themselves in heavy competition with Japanese manufacturers that were able to produce all kinds of products more quickly and of a higher quality (Hartley, 1992). To improve the competitive edge of the West there has been a lot of research on the success of the Japanese car manufacturers. Especially the study conducted by Wheelwright and Clark (1993) in the Japanese automotive industry is well known. Their study showed that Japanese car manufacturers were able to develop a new automobile in approximately 40 months while it took U.S. and Western European manufacturers approximately 60 months.

The success of the Japanese originated from their quality thinking that was based on minimising the number of defects, process orientation, and focussing on the customer using a so-called task forces. This phenomena has been studied a lot by Western countries resulting in the term Concurrent Engineering. The US Department of Defence initiated one of the major studies[8]. The study that was conducted by its Defence Advanced Research Projects Agency (DARPA), nowadays ARPA, and resulted in the first definition of Concurrent Engineering by Winner *et al.* (1988):

> "*Concurrent Engineering is a systematic approach to the integrated concurrent design of products and their related process, including manufacturing and support. This approach is intended to cause developers from the outset, to consider all elements of the product life-cycle from conception to disposal, including quality cost, schedule, and user requirements.*"

The definition stresses the integration of the different phases in the product lifecycle. Initially it only involved the overlap of product and process engineering processes. Later it also involved overlapping other phases of the product lifecycle such as operational use and service. According to Clark and Fujimoto (1991) this integration is achieved by frequent, face-to-face, bilateral communication of preliminary information instead of late release of complete information. This situation is created using multidisciplinary teams that consist of representatives from different phases of the product lifecycle. Other authors such as Carter and Baker (1992), Wheelwright and Clark (1993), and Smith (1997) refer to these task force systems as multidisciplinary teams or Design Build Teams[9].

There has been a lot of research with respect to these multidisciplinary teams. But there has been considerably less research on how to manage incomplete or preliminary information within these multidisciplinary teams. In the remainder of this chapter both will be discussed in more detail, especially the use of preliminary information because it is main topic of this research.

---

[8] Based on its first study DARPA introduced a five year program, called DICE, to encourage and enable the practice of Concurrent Engineering in the US military industry. To provide tools for the pursuit of these goals, the Concurrent Engineering Research Center (CERC) was founded at West Virginia University in the US. Today ARPA and CERC no longer focus on Concurrent Engineering research. Moreover, CE is no longer only an issue in the military world. Lots of companies in other industries are implementing Concurrent Engineering practices in their business already. A worldwide body that is still further developing and promoting the use of Concurrent Engineering is the Society for Concurrent Engineering (http://www.soce.org).

[9] Design Build Team (DBT) is a term used by Boeing (Sabbagh, 1995).

3.3.2 Multidisciplinary teams

3.3.2.1 Multidisciplinary teams versus traditional development

Prior to Concurrent Engineering the design of products was separated from the actual manufacturing of the product. The designer “throws the design over the wall” that separates design from manufacturing. A design that is the result of the "over-the-wall" principle is generally difficult and costly to produce and does not necessarily conform the desires of the market (Smith, 1997). This will result in changes late in the lifecycle, for example during production or shipping, which causes a lot of rework and delays in the project. To avoid late changes closer co-operation or integration between the different departments is required. Production, for example, should be involved in the product design as early as possible in order to make sure that the product can be manufactured in the most cost efficient way. But also other departments such as marketing, maintenance, and purchasing should be involved as early as possible.

Several levels of integration can be distinguished and the highest level of integration is a multidisciplinary team, which is also known as Concurrent Engineering. In the following several levels of integration are discussed as mentioned by Smith (1997). Firstly, integration can be realised by requiring approval by other functions. This is known as the “sign-off” co-ordination where multiple functional departments have to give their approval before a design is released to others. This is however a potentially weak form of integration because there is no mechanism to ensure that all departments are consulted in an early stage. Secondly, liaison personnel could be used. They are not members of any functional department or discipline of an organisation, but rather employees who are capable and prepared to address issues that span functional organisational boundaries. The full-time job of liaisons is to co-ordinate disparate functions. Thirdly, regular meetings (typically weekly) between employees representing different functions could be organised. During these meeting items could be discussed that are of boundary-spanning or general interest to the development project. Meetings of this type enable the organisation to maintain the functional structure while achieving a greater level of cross-functional interaction. Finally, the closest integration is realised by forming multidisciplinary teams. Under this approach, the design team is composed of representatives of several disciplines for the duration of the project. Typically such a team includes employees from design, engineering, marketing, manufacturing, and maintenance. These employees are not necessarily involved from start till finish of the project. Their involvement can vary over the different phases of a project. Often such multidisciplinary teams are located at the same location or office. This stimulates the communication between the team members, which is essential for a multidisciplinary team to be successful.

Note: The integration mechanisms that have been mentioned in this section, however, should not be confused with Design for Assembly (DFA) or Design for Manufacturing (DFM) techniques. A technique that has been developed by Boothroyd et al. (1994). The aim is to reduce the complexity of the assembly or manufacturing process by making a ‘smart’ product design. For several disciplines design rules have developed to support designers in defining product that are easy to assembly and manufacture. The rules primarily involve reducing the number of parts, simplifying the part mating and securing process, and creating symmetry or asymmetry so that it is difficult to put the parts together in any manner except the correct way (Smith, 1997). Although DFA and DFM concern the input of assembly and manufacturing knowledge, which is also an objective of the multidisciplinary team approach, it is not realised through integration but by means of design rules.

3.3.2.2 Benefits of multidisciplinary teams

There has been a lot of research with respect to the benefits of Concurrent Engineering, *i.e.* multidisciplinary teams. There are two dominant benefits, it involves the reduction in Engineering Changes and the reduction in Time-to-Market. There is a close relationship between those benefits what is illustrated using an example.

In a situation where the design is thrown "over-the-wall" to production the design is often not optimised for production. An example is that the company does not have the required equipment or tooling to realise the design. This situation leads to an Engineering Change that is fed back into design leading to a delay in the project. In a Concurrent Engineering environment production is part of the multidisciplinary team. Hence production can prevent design from making a design that cannot be produced. Because this is detected early in the process there is no need for expensive Engineering Changes and the delay in the project is reduced to a minimum.

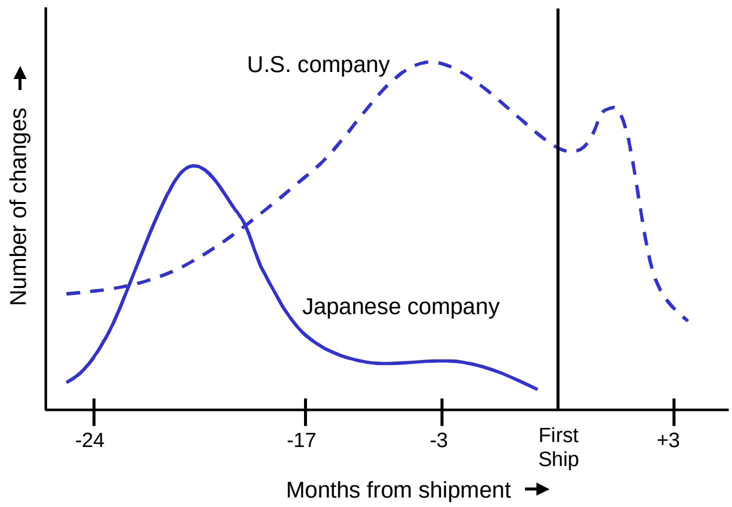


Figure 21 - Effect of CE on Engineering Changes (taken from MacKrell, 1992)

This is illustrated in Figure 21 that that is based on research that studied U.S. Companies (still using the "over-the-wall" principle) and Japanese companies (already using Concurrent Engineering) during the eighties. It shows that U.S. companies experience a lot of changes late in the lifecycle during production and even after the first shipment of the product. The Japanese on the other hand are able to concentrate the design changes during the early stages of design.

Besides the fact that late changes result in delays in the project they are also more expensive than early changes as is shown by research from DataQuest (MacKrell, 1992). The outcome of this research is that a change during design costs $1,000 and a change during final production costs $10,000,000. It is obvious that the late changes in the case of the U.S. companies are much more expensive than the early changes of the Japanese companies.

De Graaf (1996) studied several authors and provides the following summary of CE benefits.

Table 3 – Overview of benefits of CE

Issue	Metric	Results
Decreased Lead-time	Development Time	30 to 70% less
	Time to Market	20 to 90% less
Improved Quality	Engineering Changes	65 to 90% fewer
	Scrap and Re-work	up to 75% fewer
	Overall Quality	200 to 600% higher
Reduced Cost	White-collar Productivity	20 to 110% higher
	Dollar Sales	5 to 50% higher
	Return on Assets	20 to 120% higher
	Manufacturing Costs	up to 40% lower

There are not only benefits buts also risks when a company wants to implement CE practices. According De Graaf (1996) the increased product development risk is perhaps the most significant disadvantage. Because many activities occur concurrently all open issues must be resolved before the project moves into the next phase.

De Graaf (1996) also mentions known pitfalls with respect to CE as well as obstacles for further improvement. An interesting obstacle for further improvement that he mentions is the lack of information technology support. It refers to information technology that supports the sharing of information between the members of the multidisciplinary team. Electronic sharing of information makes it possible to operate as a distributed team. Hence it is no longer necessary to co-locate all the members of the multidisciplinary team to the same location.

3.3.3 Starting on preliminary information

3.3.3.1 Eppinger's framework

In a multidisciplinary team preliminary information is exchanged, which results in overlapped processes. In an overlapped process a downstream process, for example process engineering, starts before the upstream activity, for example product engineering is completed. Eppinger *et al.* (1997) developed a framework to distinguishes several types of overlap, this framework is used to define starting on preliminary information. The framework provides guidelines with respect to freezing information and exchanging the information.

The framework is based on two dimensions: (upstream information) **evolution** and (downstream information) **sensitivity**. The term evolution refers to the time that it takes that upstream generated information refines from its preliminary form to its finalised form. Thereby they distinguish between fast evolution and slow evolution. In the case of slow evolution major changes happen until late in the upstream process before which information cannot be finalised. The term sensitivity is used to refer to the impact that a change in the upstream process can have on the downstream process with respect to the moment in time that the change occurs. Thereby they distinguish between high and low sensitivity. Where high sensitivity means that a change early in the upstream process has a large impact on the downstream process and low sensitivity means that a change early in the upstream process has a small impact on the downstream process.

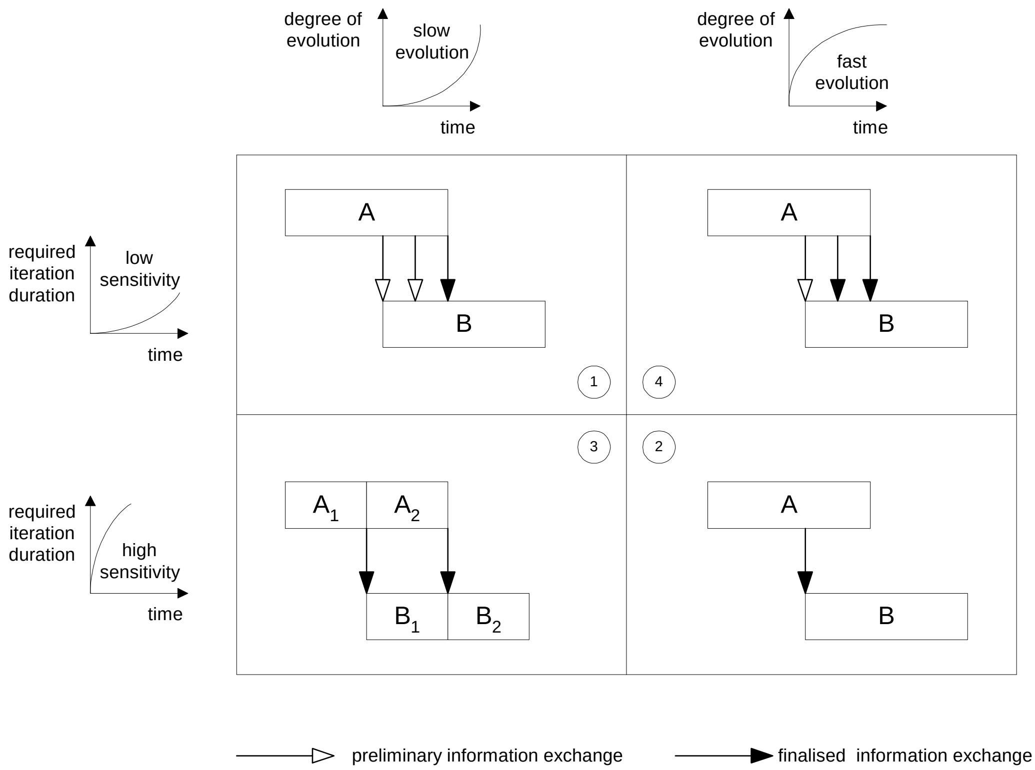


Figure 22 - Eppinger's framework

Combining these two characteristics leads to four different types of product development situations. Each situation requires a different approach with respect to overlapping upstream and downstream processes. This means that in each situation information is exchanged differently, as shown in Figure 22. The following description of the four different product development situations (or cases) is taken from an article from Eppinger *et al.* (1997).

> *Case 1:* When the sensitivity is low, it is possible to commit downstream resources based on preliminary upstream information. In this case, even if the changes in the exchanged information are large, their effects on the downstream activity are small. Furthermore, when the upstream evolution is slow - major changes happen until late into the upstream process before which information cannot be finalised - then the overlapping is said to be *iterative.* In this case, the activities are overlapped by beginning downstream activities with preliminary information, and incorporating design changes in subsequent downstream iterations**.** The design information is not finalised until the nominal completion of the upstream activity, because doing so may result in a large quality penalty for the upstream activity due to the slow evolution.
>
> *Case 2*: The opposite case is when the downstream sensitivity is *high*, but the upstream information evolution is *fast* (information can be finalised early in the upstream activity without much quality loss). In such a case, the exchanged information is to be pre-empted by taking its final value/form at an earlier point in time. In other words, the upstream problem solving is accelerated and information frozen ahead of the normal time of freeze. This is called *pre-emptive* overlapping and would help reduce development time by starting the downstream activity earlier- but with *finalised* up-stream information. Note that there are no subsequent downstream iterations. It may result in some quality loss to the upstream activity because the upstream activity loses the opportunity to make changes until its original completion time.
>
> *Case 3*: Consider the case when the downstream sensitivity is *high* and the upstream evolution is *slow.* Here it is neither desirable to start downstream activity with preliminary information nor feasible to pre-empt further changes in the exchanged information at an earlier point in time. In such a case, the exchanged information is desegregated into components to see if any of the components evolve faster or if transferring any of the components in their preliminary form to the downstream activity is practical. Often the evolution and sensitivity of the components may be different from the aggregated information (Krishnan 1996). Because the desegregation is based on the physical or functional division of the upstream and downstream activities, this approach is called divisive overlapping. If no information evolves quickly, or is it of no use to the downstream activity in preliminary form, then no overlapping is recommended with the current evolution and sensitivities.
>
> *Case 4*: The last scenario occurs when both the upstream information evolves *rapidly* and the downstream sensitivity is low. In such a case, it is possible to both start the downstream activity with advance information and to pre-empt later changes in the exchanged upstream. Because the impact of overlapping is distributed between the upstream and downstream activities (unlike in other cases), this situation is called *distributive* overlapping.

> It is worth noting that the different types of overlapping result in different tradeoffs among the performance parameters. In iterative overlapping, for instance, downstream effort is traded-off against lead-time, while in pre-emptive overlapping, upstream quality is traded-off against lead-time.

The four cases present four strategies for overlapping upstream and downstream activities. In this research the focus is on starting on preliminary information that is best described by Eppinger *et al.* as *case 1*:

> *"Starting on preliminary information is starting downstream processes before completing the upstream process by exchanging preliminary information, which is tentative, untested and possibly incorrect."*

The aim of overlapping processes is to reduce the lead-time of a project. Besides benefits there are also risks to the overlapped execution of processes. This risk is a result of iterations that occur in product development processes. As long as the iteration takes place before the preliminary information is exchanged it only affects the upstream process. However, if the iteration occurs after the preliminary information is released the upstream as well as the downstream process is affected by the iteration. In the worst case the lead-time of the overlapped process exceeds the lead-time of the sequential process. Therefore, it is important to determine beforehand whether it is worth the risk to overlap processes. According to Hanssen (2000) this is only possible if the uncertainty in the process is not too high. Otherwise it is not possible to determine the risk of overlapped processes and a company should not try to overlap processes.

To conclude this section it should be noted that Eppinger *et al.* (1997) interprets the exchange of information as the exchange of design parameters. An example of a design parameter is the diameter of an axle. In this research, however, the focus is on documents containing product definition information. In the next section it is shown how the ideas of Eppinger *et al.* can be applied to the exchange of documents in the product development process.

3.3.3.2 Eppinger's framework and the notion of documents

As described before product development is a process that creates the product definition data for production. This product definition data, for example the diameter of an axle, is often laid down in documents that are exchanged between the upstream and downstream product development activities that create and use the documents respectively. This section shows that Eppinger's framework can also be used to explain the exchange of documents between product development activities. To explain the preliminary release of documents it requires the terms document status and document versions that are introduced in section 3.2.

Case 1

In this situation the downstream activity B starts on a preliminary document version from upstream activity A[10]. As soon as the document version reaches an acceptable maturity it is released as a preliminary version to the downstream activity. There can be several iterations, resulting in new preliminary document versions that are sent to the downstream activity. Finally if activity A is completed the final document version is send to downstream activity B.

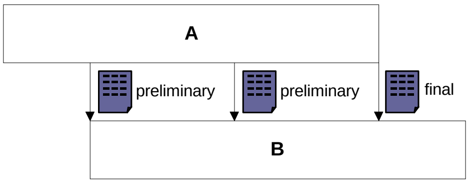


Figure 23 - Case 1: low sensitivity, slow evolution

Case 2

In this situation downstream activity B starts on a final document version of upstream activity A. A document version with status *final* is released before activity A is completed. This is possible when the document can be finalised early without much quality loss. Another possibility is that the product is over designed. This means that some safety margins have been built in. This saves time with respect to optimising the design. But on the other hand it means a higher product cost.

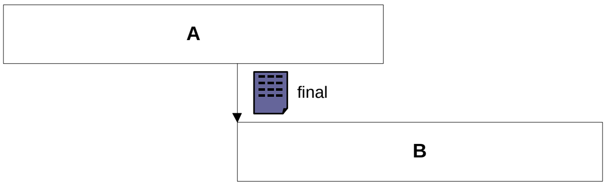


Figure 24 - Case 2: high sensitivity, fast evolution

Case 3

In this situation the document that is being exchanged is split in two separate documents and from both document on the document version with status *final* is released. This is possible if two separate activities can be identified that lead to these documents. Moreover, it should be possible to identify two separate downstream activities that need this information. An example could be separating one drawing into two individual drawings. For instance by splitting the drawing of one large facility into two smaller parts. By making two separate documents of it, it becomes possible to release one part of the facility in a first step and the other part of the facility in a second step. Hence production engineering or production can start working on one part of the facility while design is still working on the other part.

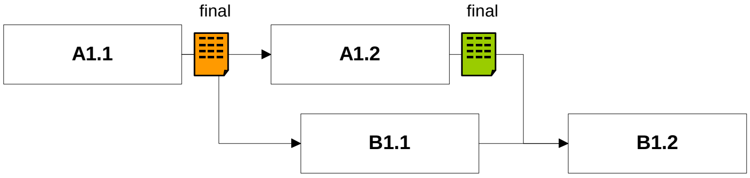


Figure 25 - Case 3: high sensitivity, slow evolution

[10] Document versions are explained in detail in section 3.2.3. For now it is enough to understand that there can be several versions of a document, which are snapshots of a document's evolution over time.

Case 4

In this situation a document version is released preliminary but soon thereafter a document version with status *final* is released. If there are any changes in the upstream activity after releasing the document version with status *final*, it results in a revised document version with status *final* that is send to the downstream activity.

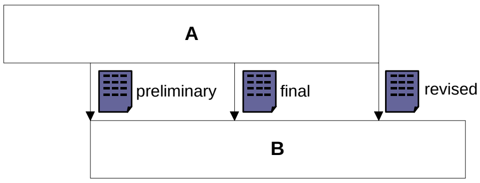


Figure 26 - Case 4: low sensitivity, fast evolution

It was already mentioned in the previous section that *case 1* represents starting on preliminary information in this research. The other forms of overlap are also important but are not part of this research. Based on this section the definition of starting on preliminary information, which was provided in the previous section, can be further refined:

> *"Starting on preliminary information is starting downstream processes before completing the upstream process by exchanging preliminary document versions, which are tentative, untested and possibly incorrect."*

Hence it is better to speak of starting on preliminary document versions instead of starting on preliminary information. In the next chapter the concept of starting on preliminary document versions is used to develop a model for product development. This model is used to further refine the definition of starting on preliminary document versions.

4 Research model: Product Development Process

This chapter presents a research model that models product development as a network of related document lifecycles. This model helped to develop my theory on the enabling effect of product data management with respect to starting on preliminary information, which is presented in the next chapter.

4.1 Introduction

Models of product development can be found in literature on Design Methodology. De Graaf (1996), Roozenburg and Eekels (1998), Erens (1996), and Dorst (1997) each identified several classes of product development models. There is much overlap in the classes therefore their views are combined into the following overview of classes of product development models.

1. **Descriptive Models** or **General Problem Solving Models**. These models describe the complex behaviours of human beings that are required to solve engineering problems in terms of underlying functional mechanisms (Erens, 1996). In other words they view the product development process as a problem solving process. The model of French (1985) is an example of such a model and seems to be valid in general. De Graaf (1996) mentions that the major drawback of these models is that they suggest that the development process is heuristic, which cannot provide a guaranteed success when used.

2. **Prescriptive models** or **Phase Models**. The prescriptive models are typical examples of technical positivistic models, which focus on the procedures and strategies for executing the product development process. They often suggest a systematic approach such as the standard design process (VDI-2221) developed by the German Society of Engineers (VDI). If designers follow these (standard) approaches it will lead to better designs.
   Pahl & Beitz (1984) and Van den Kroonenberg & Siers (1993) have developed similar models. Appendix 1 shows a comparison of the three descriptive models that are mentioned. The comparison reveals that these models recognise similar stages in the product development process.

3. **Management Models**. These models focus on the management aspect of product development activities. An example is the Integrated Product Development model from Andreasen & Hein (1987) in which they link the management and execution of Product Development activities. Ullrich & Eppinger (1995) also accentuate the management's role in Product Development.

4. **Artefact models**. According to Erens (1996) artefact models are design methods that prescribe the design process by prescribing the attributes of the artefact. They are closely related to product modelling languages with the objective of supporting decisions in the design process. An example of an artefact model is Axiomatic Design (Suh, 1990).

However, none of these models focuses on the release process of documents in the product development process, which is relevant for studying starting on preliminary information [11]. Therefore, a new model of the product development process is developed in this research that

[11] N.B. Literature on starting on preliminary information is limited in general

focuses on the release process of documents. Although none of the mentioned models focuses on starting on preliminary information it was believed that a slightly adapted Axiomatic Design model could be used as a starting point. In the next paragraph Axiomatic Design is discussed in more detail. It is shown how the Axiomatic Design model can be extended so that it can be used to describe the release process of documents in the product development process.

4.2 Axiomatic Design

4.2.1 Notion of domains

Axiomatic Design is a model of product development that was introduced by Albano *et al.* (1993) and Suh (1990). It focuses on the artefact that is being developed, *i.e.* the aircraft, automobile, shaver, or CD player[12]. Axiomatic Design discusses product development in terms of the evolution of the artefact. This evolution takes place in several domains where each domain has its own product model of the artefact (see Figure 27).

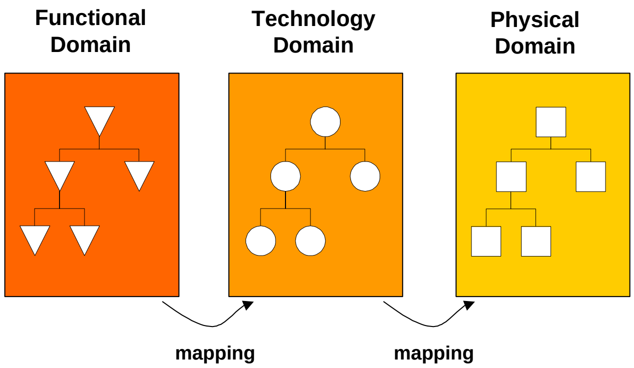


Figure 27 - Domain model of Axiomatic Design

Albano *et al.* (1993) identified three different domains: Functional Domain, Technology Domain and the Physical Domain. Hence they also identified three models of the artefact: Functional model, Technology model and a Physical model. Looking at product development from a Axiomatic Design point of view, the activities in the product development process have to create the product models in the different domains.

The domains in Axiomatic Design are ordered. This order says something about the sequence in which product development activities are executed. Product development starts with developing the functional model. The functions in the functional model are then mapped onto the technology model. The mapping process means that it determines which technology can fulfil one or more of the functions. The technology model is in turn mapped onto the physical model, which consists of components, which fulfil one or more technologies.

The idea behind Axiomatic Design is the fact that is very hard to derive the physical model, *i.e.* assemblies and components, directly from the requirements. Especially if it involves

---

[12] Also authors use the notion of domains, Andreasen & Hein (1987) for example defined similar domains in his so-called chromosome model. The idea of domains is also used in the widely accepted model by Pahl & Beitz (1984). In their model they distinguish three major product models: functions, solution principles and product modules.

complex products, where several intermediate steps are required. The domains and the corresponding product models represent these intermediate steps.
There are not always three intermediate steps or domains although this is suggested by the model of Albano *et al.*. It depends on the company how many domains and hence product models that can be identified. A craftsman for example that designs a chair does not create a functional and a technology model. He directly starts with making a design of the chair, *i.e.* a drawing in the physical domain. Erens (1996) mentions that explicitly modelling the different product models (in different domains) is only required when it involves complex projects in which hundreds of designers are involved.

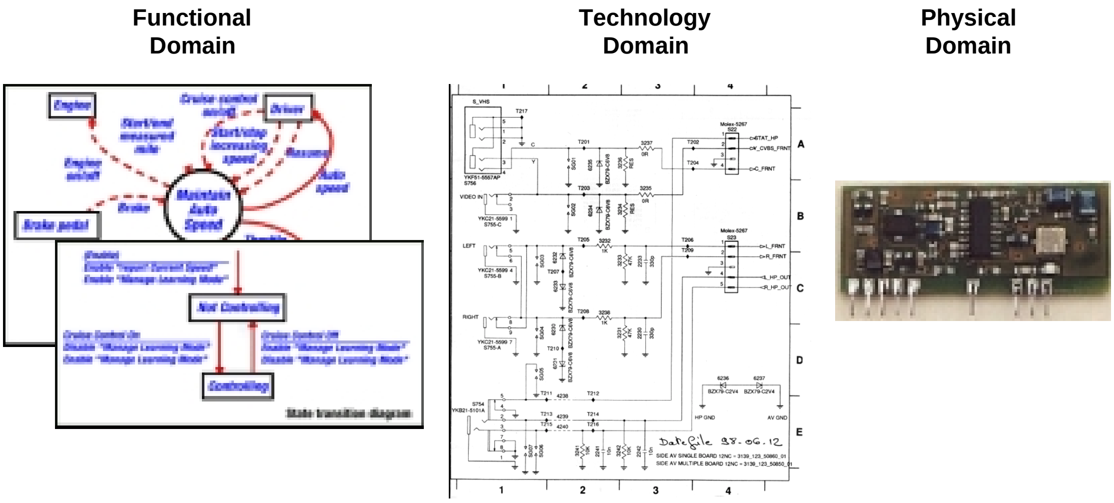


Figure 28 - Example of representation of product models in different domains

The different domains and product models of Axiomatic Design are illustrated using an example derived from PCB design (see Figure 28). The process starts with listing the requirements for the PCB in a text document. These requirements are mapped onto the functional product model. The functions of the PCB are modelled using a modelling language for example input/output diagrams or state transition diagram. The functional model is translated into a technology model. The technology is modelled using a circuit diagram, which is a logical diagram of the PCB. Finally the technology model is mapped onto the physical model, which is modelled using a E-CAD application to model the exact location (PCB layout) of components on the board including the wiring.
The mapping process results in relationships between functions and technical solutions and between technical solutions and the physical model. It is important to keep a record of these relationships in case there is a change made for instance to a function. The relationships then identify which technologies or even physical components are affected by the change.

4.2.2 Extension of Axiomatic Design

As already mentioned in the introduction, none of the models focus on the release process of documents. This is also true for the Axiomatic Design model, which does not even contain the notion of documents. In this section it is shown how the notion of documents can be introduced to the concept of Axiomatic Design.
In the previous section the product development process is described in terms of domains and product models. These product models are represented as tree structures. The functional model, for instance, can be represented by a tree consisting of functions, sub-functions, and

sub-sub-functions. According to Erens (1996) the components of the tree structures are described using different modelling languages. A description of a component using a certain modelling language is called a representation. These representations are stored in a document such as a text document or a drawing. Humans who understand the semantics and syntax of these representations can interpret the contents of these documents.

This is illustrated in Figure 29, which shows an example of the product tree in the Technology domain, in this example the artefact is an X-ray system[13]. It shows that there is a corresponding document for every part of the product model that describes the particular part. On the lowest level in the hierarchy of the product model individual parts are described while at higher levels the interfaces between the parts of lower levels are defined. Examples are mono drawings that describe parts and assembly drawings that describe the interfaces between these parts.

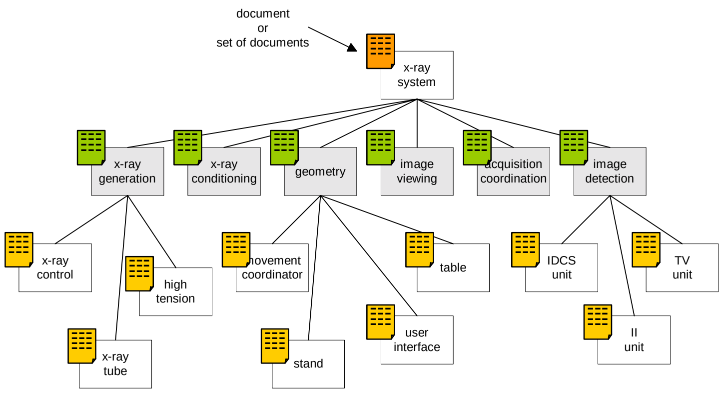


Figure 29 - Product model in technology domain and its related documents

In an ideal situation each component is described in a different document. However, it is often seen that for reasons of maintenance or overview the author prefers a document that covers several components (Erens, 1996). An example is combining the mono drawings of several small components into one large drawing.
Moreover, a component is not necessarily described by one document. In the physical domain for example there might be a drawing of a part as well as a calculation that proves that the part on the drawing is strong enough. But for the sake of simplicity it is assumed that a document contains the representation of only one component of the product model and that there is only one document for each component.

If the above is applied to the product development process it should be possible to identify all documents that are created in a product development process. Of course it only involves documents that define the product and it does not involve documents such as memo's, e-mails, minutes of meeting etc.
In practice it means that first the domains should be identified that are used in the product development process. Consequently this leads to the identification of the product models, *i.e.* one product model for each domain. Decomposing each product model results in a tree

[13] Example derived from Erens (1996).

structure that leads to the identification of (abstract) documents that are created in the product development process[14].

This concept of decomposition does not cover every design activity in the product development process. Its focus is on documents, which is important for my theory on starting on preliminary information. The next section focuses on the document release step in the product development process, in this section the concept of decomposition is further detailed.

4.3 Model of Product Development

In this chapter a model is presented that models the product development process as a process, which creates and releases documents. An important aspect of this model is the relation between the release of individual documents. This relation determines the sequence in which documents are released.

The model that is created is on the (abstract) document level; the use of document versions is explicitly mentioned in this section.

4.3.1 Introduction

As mentioned before the extended Axiomatic Design model that was presented in section 4.2.2 can be used to identify all documents that are required to define a product. From a document point of view the product development process can be modelled as the process that creates and releases all these documents. The question is, however, in which sequence the documents are created and released?

The sequence or precedence relationships are determined by the input and output relationships between documents. A document that is created serves as an input for another document. By analysing a product development process one could identify all the input and output relationships between the documents that are created and released in that product development process.

However, instead of analysing each individual product development process one could also apply general rules concerning the precedence relationships of documents in the product development process. This assumption is based on the fact that Erens (1996) identified four elementary mechanisms that are used in design: *decomposition*, *allocation*, *composition*, and *validation*. According to Erens (1996) these mechanisms seem to be generally valid in design and engineering environments across industries. Designers across industries use these 4 mechanisms to develop new products.

Together these 4 mechanisms constitute Erens' Design Cycle (1996) that is based on March's Productive Reasoning model (1984). Figure 30 illustrates the Design Cycle concept showing the relation between the four elementary mechanisms. In this figure the first design step is *decomposition* and refers to the fact that an overall problem is decomposed into 'easier' (less complex) sub-problems that can be solved separately. *Allocation* corresponds to solution search and exploration for solutions that are found for the different sub-problems. This can be an iterative process, which means that solution search and exploration can lead to a change in the overall problem. Moreover, they are looking for several alternatives to solve the problem so that they can select the best alternative.

[14] All identified documents are not necessarily created. When complete modules are re-used the documents that describe the modules are re-used as well as the documents that describe the corresponding technologies and functions.

Composition is the act of constructing the overall solution from a related set of sub-solutions. The allocation results in sub-solutions for the sub-problems. These sub-solutions have to be integrated into an overall solution that works. The last step is *validation* in which it is verified whether the overall solution is in line with the overall problem. In other words does the overall solution still matches the requirements.

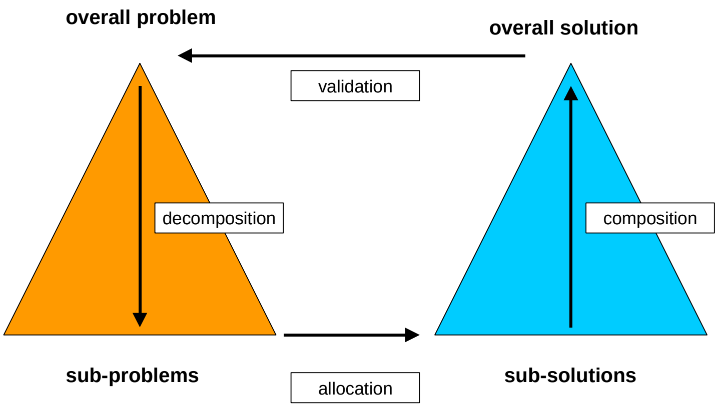


Figure 30 - Design Cycle of Erens (1984)

In his thesis Erens also shows how the Design Cycle can be applied to the Axiomatic Design model. The decomposition and composition mechanisms are applied to the product model in a domain. While the allocation and validation steps apply to the product models in different domains. In the next sections it is shown how the (extended) Axiomatic Design and Design Cycle model can be used as a guide to identify precedence relationships between documents.
First the sequence in a sequential product development process is discussed. This is a situation that is still found in companies today. After that the sequence in a concurrent product development process is discussed. By comparing this situation to the sequential product development process it is easier to understand what problems companies experience that want to make the shift from a sequential to a concurrent product development process. This help in defining my theory on the enabling effect of PDM with respect to starting on preliminary information.

4.3.2 Sequential product development

In this section a sequential product development process is discussed using the extended Axiomatic Design model and the Design Cycle model. As an example an imaginary company is used with a product development process that consists of two domains, *e.g.* a functional and a physical domain.
In a sequential product development process all documents in the functional domain should be released before creating documents in the physical domain. From a project management view a sequential product development process is a process that is divided into several phases that are executed sequentially. Applying this to the above each domain could be a phase in the product development process, for example the functional design phase and the physical design phase.

Applying Erens' Design Cycle shows how an ideal sequential product development process would be executed. During the functional design phase the problem is decomposed into smaller sub-problems resulting in a function tree (decomposition mechanism). This top-down approach suggests that the document describing the main function should be released before one can start working on the documents describing the sub-functions. In other words there is a so-called *Finish-Start* relationship between lifecycles of these documents. Because it involves a relationship between documents of the same domain it is called an *intradomain* relationship.

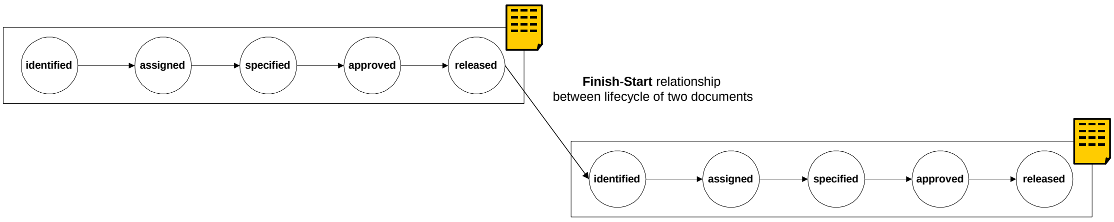


Figure 31 - Finish - start relationships between documents

Figure 31 shows an example of a Finish - Start relationship between two (abstract) document lifecycles. The document lifecycle, as presented in section 3.2.4.3, is used as an example. The figure shows that there is a Finish-Start relationship between the release status of one document and the identified status of the other document. Because of this relationship the documents are created and released sequentially.

Once all documents of the functional design phase are released one can start with the physical design phase. During the physical design phase the functional components on the lowest level of the function tree are mapped onto physical components that realise the function (allocation mechanism). In other words there is precedence relationship between these documents that is called a Finish-Start *interdomain* relationship.
The individual sub-solutions are integrated into an overall solution (composition mechanism). This bottom-up approach also suggests that first the description of the physical components must be released before one can start with the creation of the assembly of these components. This is also a *Finish-Start* intradomain relationship.

All documents of the physical design phase are not released to production before the overall solution in the physical domain is validated against the overall problem in the functional domain (validation mechanism). After validation all documents in the physical domain are handed-over or released to production.

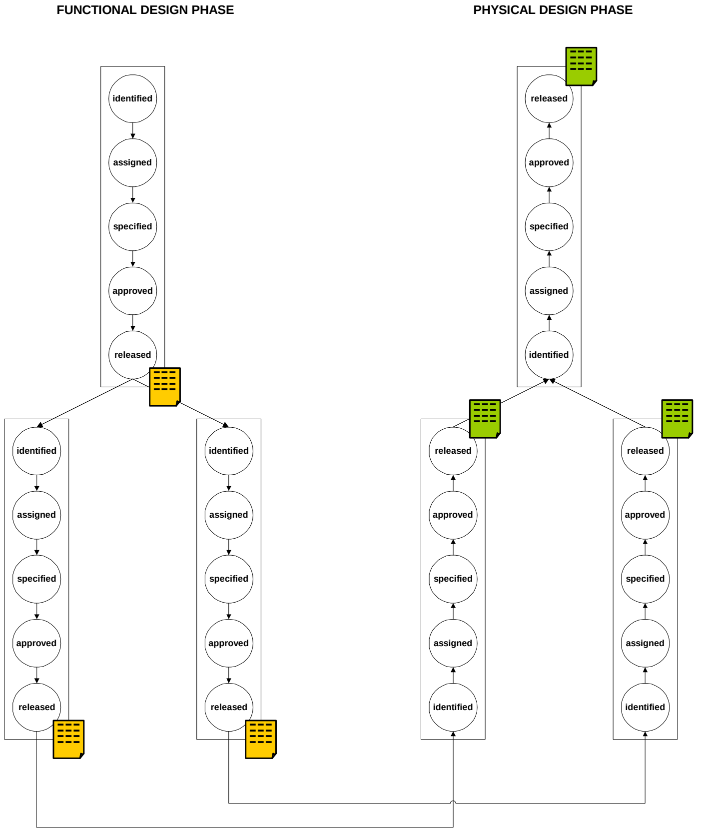


Figure 32 - Finish - start relationships in sequential product development

The ideal sequential product development process is illustrated in Figure 32. In this figure it is shown that the document lifecycles of the individual documents are connected by Finish-Start relationships and are therefore, executed sequentially. The figure also shows that documents on the same level in the tree structure representing the product model are executed in parallel. This is of course only possible when there is a good interface definition available and when there are enough resources to work on both documents in parallel.

4.3.3 Iterations in product development

Although it was not discussed in the previous section, there are iterations in each product development process. These iterations result in document versions and make the product development process more complex, because it is difficult to predict the number and length of each iteration. In this section three types of iterations and their impact are discussed. Moreover, a discussion is presented which addresses the need for a structured release and version control process to do controlled iterations.

The three types of iteration that are distinguished concern:
- Creative iteration
- Intradomain iteration
- Interdomain iteration

These three types of iterations are schematically shown in Figure 33 and will be discussed separately.

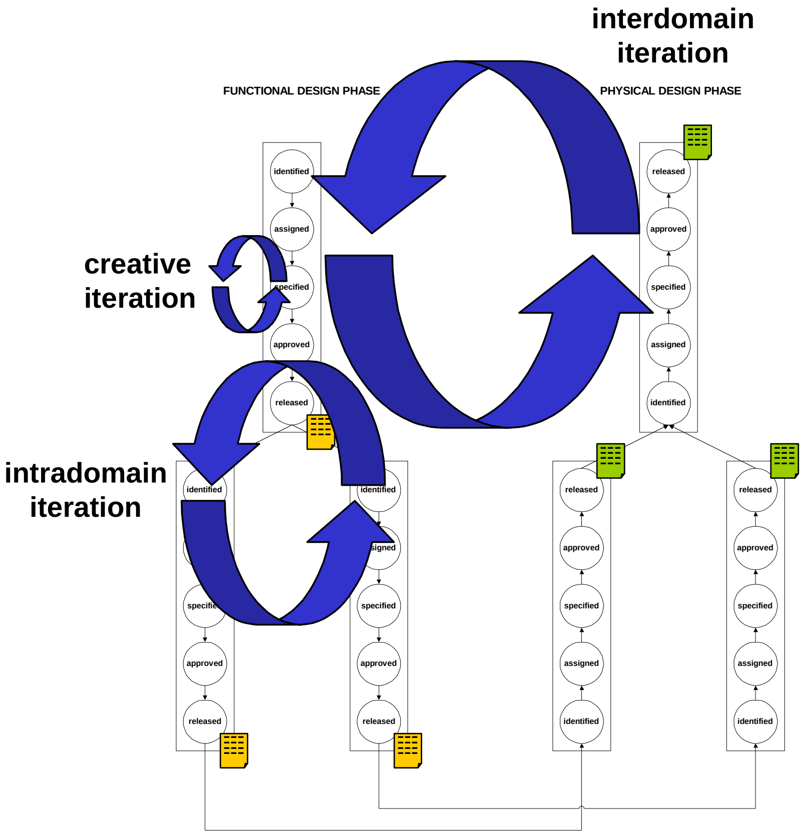


Figure 33 - Three different types of iteration

Creative iteration

This type of iteration is described by Clausing (1994) and Roozenburg and Eekels (1995). Clausing (1994) writes:

> *"We put a design on paper, and then consider it from many perspectives. The increased understanding leads us to the modification of the design. This iteration is inherent in human limitations. We cannot conceive an idea, consider it from all perspectives instantaneously, and draw the final design … Thus we must iterate."*

Roozenburg and Eekels (1995) more precisely describe the human limitations:

> *"Designing is a process in which all sort of things are done (drawing, building models, experimenting etc.), but above all, it is a process of goal oriented reasoning. The reasoning from function to form is a form of deductive reasoning. This means that the conclusion (the design) does not indisputably follow from the premises (the functions to be fulfilled), and that, in principle, there are many good solutions. The reasoning from function to form is a creative process, which can be encouraged methodically, but cannot be logically guaranteed."*

Obviously several iterations are needed before a solution is satisfying and meets the requirements. These iterations often result in several possible solutions from which the best alternative is selected that is worked out in detail and is finally released. In terms of a document lifecycle the creative iterations take place in the edit phase before the document is released.
From the above it can be derived that a document should not be released before the creative process/iterations are ended. If a document is released before this creative process is ended it leads to changes in the released document and hence re-work and delays.

Intradomain iteration
The intradomain iteration applies to two or more documents in the same domain that have a precedence relationship. This is illustrated using the example of an assembly that consists of several mono parts in the physical domain. In this example the assembly is represented by an assembly drawing and each mono part is represented by a mono drawing. Moreover, it assumed that the assembly drawing is created and released before the mono drawings are created and released.
While working on the mono drawings the designer might discover an error. It might for instance be impossible to make a part that fits into the assembly. This has consequences for the design of the assembly and hence for the assembly drawing. As the assembly drawing is already released a new version of the assembly drawing has to be created and released. Such a change is also an iteration and is called an intradomain iteration.

Interdomain iteration
The interdomain iteration applies to documents in different domains. This can be illustrated using the same example of the assembly that consists of several mono parts in the physical domain. Once again while working on the mono drawings the designer might discover an error.
This time the designer discovers that it is impossible to define a part that has the functional capabilities as described in the functional domain. This has consequences for the functional description. Because the functional description has been released it results in a new version of the functional description. Such a change is also an iteration that is called an interdomain iteration.

Note: Working in multi-disciplinary teams can prevent such type of iterations. Because of the communication in these multi-disciplinary teams you already find out that it is impossible to realise a function because the people that have to make the physical design are involved in the team and can provide feedback before the functional description is released.

It is impossible to prevent iterations taking place. Therefore, a working practice is need that is able to deal with these iterations in a controlled manner. A well-defined release and version control process is such a working practice. As mentioned before documents should not be released, *i.e.* frozen, before the creative iterations are ended. Once the creative process is ended that document version is frozen so that it forms a stable input for further design activities.

After the document version is frozen it is principally not allowed to make changes to the document version anymore. Changing the document version would namely lead to iteration in the documents that use that document version as input; this is called propagation of changes. All these changes mean extra cost because activities have to be re-done. Moreover, iteration leads to delays in the product development process.

However, as we already mentioned it is impossible to prevent iterations. If an intradomain or extradomain iteration is required once a document version is frozen it leads to a change proposal. This means that the cost and benefits of this change are assessed. If the benefits outweigh the cost the change is made to all documents that are involved. This procedure is a controlled manner of conducting iterations.

In the next section it is shown that the release process that is used for a sequential product development process is not suitable for a concurrent product development process. Moreover, it is shown that in a concurrent product development process it is even more important to have working practices to deal with iterations in a controlled manner.

4.3.4 Concurrent product development

The previous section showed that the two phases, which in this case match with two domains, are executed completely sequentially. Meaning that all documents from the functional design phase are not handed over to the physical design phase before the end of the functional design phase.

4.3.4.1 Overlapping phases to start on preliminary information

There are several ways to introduce overlapped phases in a sequential product development, which are discussed in this section. However, there is only one way to overlap phases by starting on preliminary document versions.

Firstly, overlapping phases can be achieved by releasing a document as soon as it is completed. Therefore instead of releasing all documents of a phase at once, each document is released individually. Reinertsen (1997) already described this principle. He mentions that from a queuing theory point of view it is better to release documents in small batches (*i.e.* individually) instead of large batches (*i.e.* all at the same time). This reduces the waiting time and hence shortens the throughput time of the individual documents.

However, if documents are released to the next phase when they are ready they lack a consistency check. For example, all mono drawings of the same assembly should be checked for consistency to make sure that the parts fit together. In a sequential process such a check is normally done at the end of a phase where a check is conducted to ensure all the documents are consistent. When documents are already released before this final consistency check one could call this a form of starting on preliminary information.

In terms of Eppinger’s framework such a release process is a Case 3 (see section 3.3.3.2). The total set of documents that is released at the end of a phase is broken down into smaller parts, *i.e.* individual documents, which are released individually. In this research starting on preliminary information is defined as Case 1 of Eppinger’s framework. Documents are issued (*i.e.* distributed) before they are finally released. Two variants are distinguished, variant 1 and variant 2.

Variant 1
Variant 1 is explained using the document lifecycle from section 3.2.4.3. Assume that there are two documents A and B, both having status identified, and that document B needs document A as input. It is possible to define that a version of document A can be issued to be used as input for document B before document A receives status released (see Figure 34).

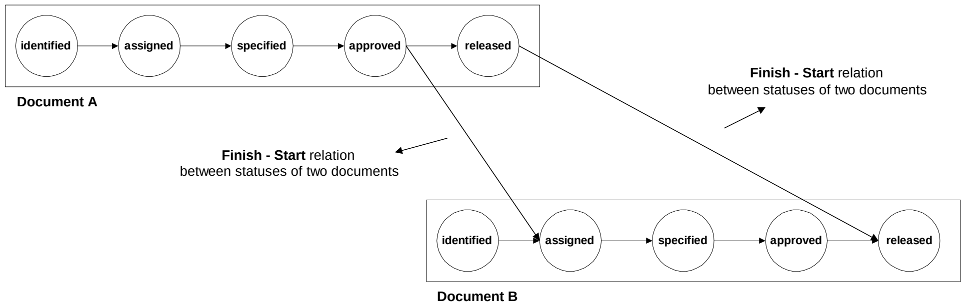


Figure 34 - Preliminary release of a document version

In Figure 34 it is shown how document B starts on a preliminary version of document A (it is called preliminary because it does not have status released). More precisely it shows that document B can be assigned to an engineer (*i.e.* status assigned) when a version of document A is approved by the reviewers (*i.e.* status approved). The release of a preliminary document version is modelled as a Start-Finish relationship between lifecycle statuses of the individual documents. Figure 34 also shows another Start-Finish relationship, namely between the released statuses of documents A and B. This relationship indicates that document B might not be released before document A is released.
In variant 1 it is important that a document is released preliminary <u>after</u> it has reached status specified. This is important because once the document has reached status specified the designer/engineer is no longer working on the document. Hence it is possible to use the 'traditional' document lifecycle in the release process.

N.B. The preliminary release process that has been described is actually used by a manufacturer of copiers (Pels *et al.*, 1995).

Variant 2
Variant 1, however, does not always lead to significant reductions in throughput time. The reason is that it can take weeks to specify a document and it takes only a few days to approve and release the document. In order to further reduce the throughput time the document should already be released preliminarily while the author is still specifying it. This is called variant 2 in this research.
Basically companies already do this. If somebody needs a preliminary version of document A before it is specified he asks the author of document A whether he can receive a preliminary version. The person that submits the request indicates for what purpose he needs a preliminary version of the document. The creator will determine whether the quality or maturity of the document is sufficient for the indicated purpose. If the creator agrees he will create a document version with status *assigned* (what basically means *in work*) and a banner/watermark printed on each page that says "FOR INFORMATION ONLY". It is

important to notice that the content of the document in this example is not frozen. Therefore, the person that receives the preliminary version has no guarantee that there will be no more changes.

In the previous section it was mentioned that it is important to have a controlled release process because of the iterations. The example that was just described deals with interdomain and intradomain iterations but also with creative iterations. Moreover, people do not work with frozen documents anymore, which further increases the risk of iterations. This problem can be solved by using a more formal procedure for preliminarily releasing documents.
Hanssen (2000) already mentioned that starting on preliminary information should be done when it is possible to:

- define the content of information (*i.e.* document) that is released preliminary,
- define the point in time that the preliminary information (*i.e.* document) is available, and
- define the decisions that can be taken on the basis of this preliminary information (*i.e.* document).

This means that companies have to adapt their release process. To create a more formal release process that supports the release of preliminary documents, the evolution of the content of a document (*i.e.* its maturity) should be expressed in a document status. It should be clear to everybody in the organisation what a specific status means and which decisions can be taken on the basis of a document with that specific status. This results in the following definition of preliminary information.

Preliminary information:
A preliminary document version whose preliminary form is expressed in a document status. If the document version reaches that status it is preliminarily released (verified and authorised) for a specific purpose to a specific audience. The purpose indicates which decisions the audience can make on the basis of the preliminary document version.

One could define that a document is released preliminarily several times during the process of creating the document. This means that a document can have different release steps and that every release step has a different purpose. According to Hanssen it should be possible to define the content of the preliminary document version, the point in time that it is preliminarily released, and the decisions that can be taken (*i.e.* the purpose) on the basis of the preliminary document version.
If this is possible the preliminary release steps can be modelled in the document lifecycle. The purpose of the preliminarily release of a document should then be expressed in a status. For example, if a document is released preliminarily because Purchasing needs dimensions of a plate to order material, the status of the released document version could be Released for Purchasing, for example.
In order to control the risk of iterations it is important that there is some kind of version control applied to the preliminary released document version. Moreover, the receiver of the preliminarily released document version should have some kind of guarantee that the content of the document version is validated and will not change anymore.
Therefore, every release step requires verification and authorisation because others will use the information. Therefore, it should be verified that the information is indeed ready to be Released for Purchasing. By authorising the release of a preliminary document version the

receiving party knows that it can use the preliminary document version to take some pre-defined decisions, in this case ordering the plates.

The preliminary release means that the document is partly frozen. In the example it means that the creator is free to change everything except for the dimensions of the product because they have been preliminary released to Purchasing. If a change is required in the dimensions there should be a change procedure to ask permission to change the dimensions that were already preliminary released. By freezing the dimensions the Purchasing department has the guarantee that it can order the plate with very low risk. By applying some safety margin, *e.g.* a plate with width 600 mm instead of 570 mm, one still has the possibility to cope with small changes.

4.3.4.2 Document workflow model including preliminary release

The traditional release process is not capable of dealing with the preliminary release of document versions. The major difference is that in the traditional release process each version goes through the same document lifecycle. In a preliminary release process, as presented in this section, document versions can go through different lifecycles. This is explained using the document workflow model that was introduced in section 3.2.4.4 (Figure 35).

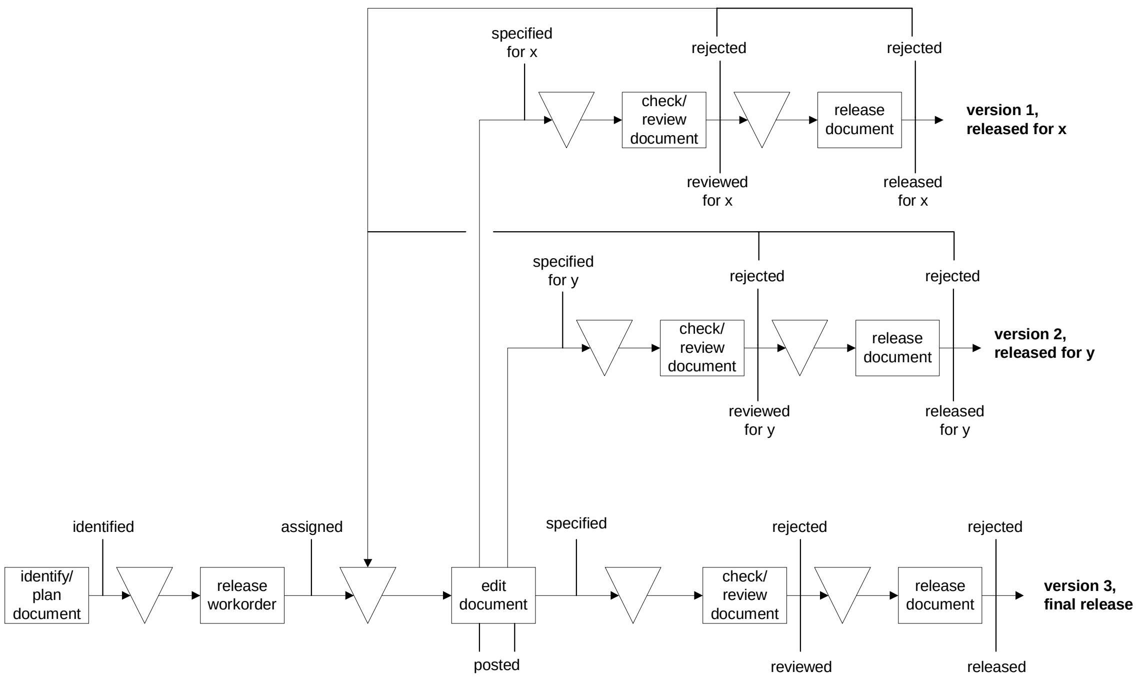


Figure 35 - Document workflow model including preliminary release

The document workflow model shows that the first document version is identified and assigned to a designer/engineer. Then the designer/engineer starts working on the first document version. When the document reaches certain maturity a preliminary document version, version 1, is *released for x* (Figure 35). The engineer continues working on the preliminary document version (that is partly frozen) by creating a new document version, version 2. After reaching a certain higher maturity level the second preliminary version of the document is *released for y*. Once again the designer continues working on the preliminary document version by creating a new document version, version 3. When the document is completed, version 3 it is submitted for the final release. This example shows that different document versions of the same (abstract document) can have a different release process.

4.3.4.3 Status - version matrix of preliminary release process

The preliminary release process of a document illustrated using a status version matrix, which was introduced in section 3.2.4.5. The status - version matrix is briefly discussed in this section using Figure 36.

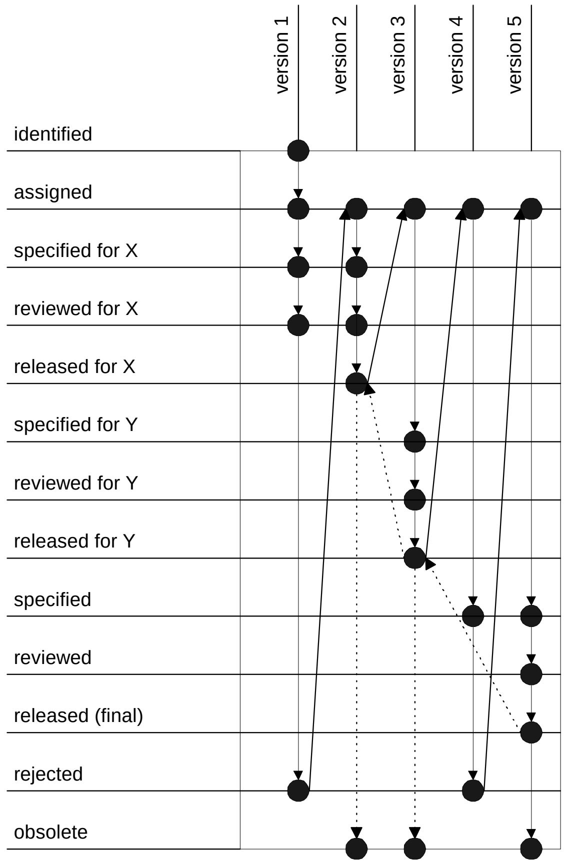


Figure 36 - Status - version matrix including preliminary release

Document version 1 successfully goes through the first three phases, but it is rejected during the approval step. This results in an iteration and the creation of document version 2. Document version 2 is *Reviewed for X* and is also *Released for X*. Hence others can use this preliminary document version as input for their (x-)activities[15]. But the document is not complete yet so the creator continues working on the document by creating a new document version. Document version 3 is successfully submitted for review and receives status *Reviewed for Y* and *Released for Y*. Document version 3 with status *Released for Y* replaces document version 2 with status *Released for X* which receives status *Obsolete* now (see dotted arrows). Still the document is not complete yet and the creator continues working by creating a new document version. Document version 4 is not approved during the review resulting in the creation of document version 5 that successfully goes through all statuses and finally

[15] As mentioned before it should be (pre)defined who can use a preliminary released document and for what purpose. The purpose is expressed in the status of the document version. In this example an imaginary purpose is used, i.e. X and Y.

receives status *Released*. Document version 5 does is not a preliminary version anymore, but the final version because the authors considers the document to be complete.

4.3.4.4 Model of concurrent product development process

Using the document lifecycle model for preliminary releasing document versions and the extended Axiomatic Design model it is possible to describe a concurrent product development process. An example is shown is Figure 37 using a simplified lifecycle model, in which some statuses are omitted for the sake of simplicity.

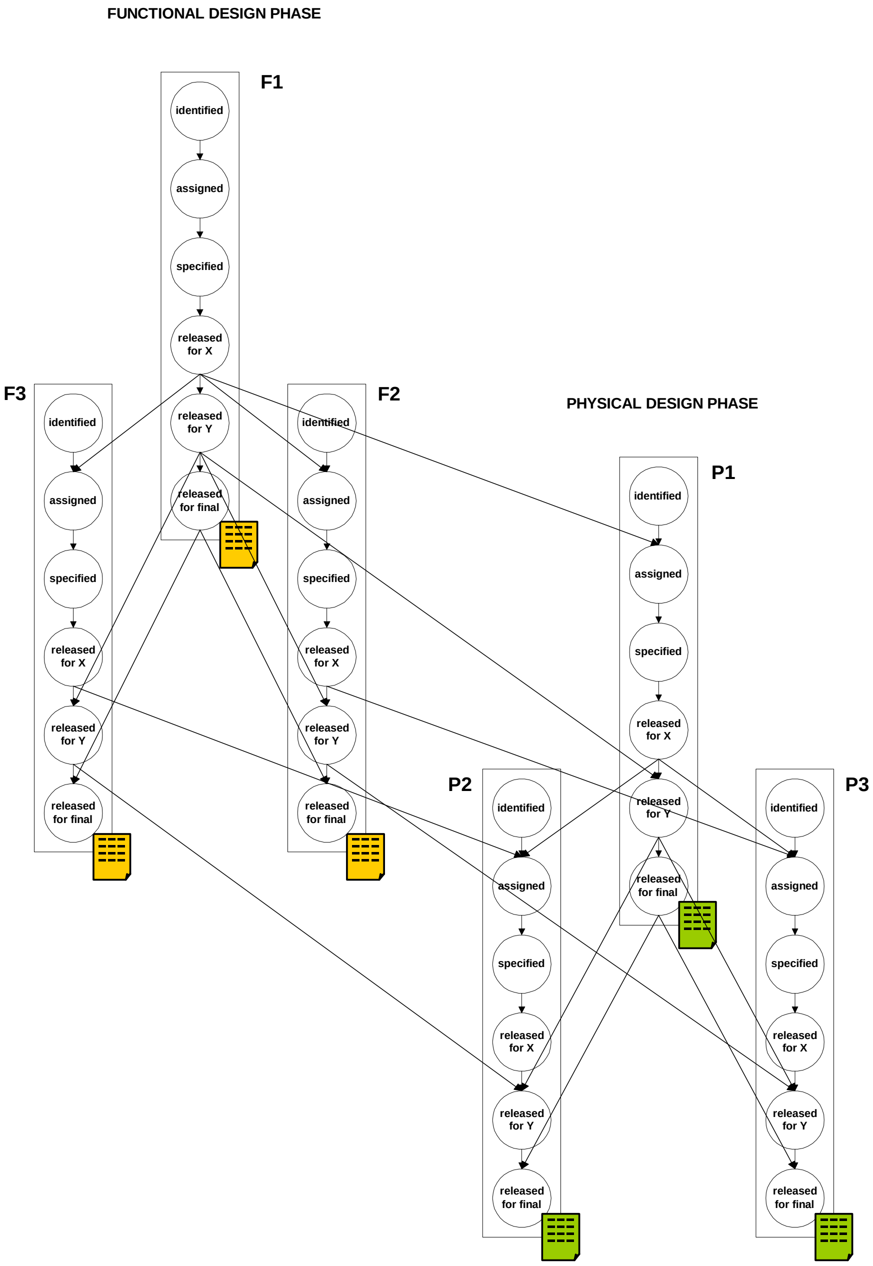


Figure 37 - Concurrent product development using preliminary release of documents

Figure 37 shows that the product development process starts by working on one document in the Functional Design Phase (document F1). This preliminary document version is released so

that others can start working early on the preceding documents (F1 and F2) in the Functional Design Phase. This is represented by the *Finish-Start* relationship between the statuses of the document lifecycles of F1 and F2/F3. It concerns a so-called *intradomain* Finish-Start relationship.
Moreover, they can also start with the first document in the Physical Design Phase (P1). Once again there is a Finish-Start relationship but this time it concerns an *interdomain* Finish-Start relationship. Because of the release of a preliminary document version of document F1, documents in the Functional Design Phase as well as the Physical Design Phase are in progress, in other words both phases are overlapped.
There are also relationships that prevent the preceding documents having a higher status then the document that serves as input. Document F2 and F3, for example, can not have status Released for Y before document F has reached status Released for Y. This prevents too much work having to be re-done in the case of an iteration.

The overlapped process as presented in Figure 37 results in a complex network of intradomain and interdomain Finish-Start relationships. As mentioned before Figure 37 is just an example, companies can define their own preliminary document statuses and how they connect the document status of preceding documents using Finish-Start relationships.

4.4 Increased control complexity

In the previous section it was discussed how starting on preliminary information can be used to overlap phases of a product development process. As briefly mentioned, it results in an increased control complexity of the design release process. This is discussed in more detail in this section.

In the traditional product development process the complete design is released at the end of a phase. This means that all documents that are related to the design are released at the same time. In the previous section it was shown how phases of the product development process are overlapped by releasing individual documents or small sets of documents instead of one large batch of documents.
This affects the control complexity of the product development process. It changed from managing one large batch of documents to be available at the same point in time, to managing small sets of documents to be available at different points in time. It is assumed that this requires that each small set of documents is monitored and tracked individually.
This implies that information is gathered about the progress of each set of documents. This makes it possible to compare the progress of a set of documents with its scheduled release date. After all, the preliminary status tells you something about the maturity of the document and therefore, tells you something about progress of the activity that creates the document.

But there is an additional factor that further increases the control complexity. Small sets of documents or single documents are released in preliminary form instead of in final form. This requires the use of a release process that includes the release of preliminary document versions as discussed in section 4.3.4. Moreover, it increases the number of releases that have to take place. In the traditional situation there is only one release, the final release, and in the preliminary situation there can be several preliminary releases before the final release.
This requires extra attention in terms of monitoring and tracking the creation of documents because of the increased risk. Releasing preliminary versions of documents increases the chances of re-work and delays, because there is a higher chance for changes and thus iterations. If there is a change in a document from which a version is preliminary released to a

downstream process, there is a risk that the change will affect the activities in the downstream process that started on the basis of the preliminary released version of the document.
Moreover, there is a risk that a change results in an avalanche of changes because more than one document can be affected by the change. This effect is called propagation of errors or changes. Normally a change, *e.g.* a change in material, affects a component or assembly of the product. In such a case it should be determined whether the change only affects the component from which the material changes. If the component is part of an assembly the change can also affect other parts of the assembly. In other words the change propagates from one component to the other.
The consequence of a change is that documents describing the parts need to be updated. If a preliminary version of these documents already has been released it means that it also affects activities in downstream processes. The change propagates from one activity to another. For example, assume that activity A provides input for activity B and that activity B provides input for activity C. Moreover assume that activity B starts on a preliminary version of the document produced by activity A and that activity C starts on a preliminary version of the document produced by activity B. If there is a change in the preliminary released version of the document produced by activity A it affects activity B as well as activity C.

As a change can cause a lot of re-work two things are important:
- the impact of a change should be determined quickly in terms of affected components and documents, and
- the persons that started on preliminary versions of the affected documents should be notified quickly.

If this is not done adequately employees will continue to work on (preliminary) document versions that become outdated. This will result in extra re-work, which in turn result in a delay of the project. However, it should be noted that the project is only delayed if the activity is on the project's critical path. In case an activity is not on the critical path it will only result in extra cost because of re-work.
The following text box provides a summary of this section. In the next section it is discussed which PDM functions are required to manage the control complexity that is introduced by starting on preliminary versions of documents.

Starting on preliminary information increases the control complexity because:
- smaller sets of documents have to be monitored and tracked,
- these smaller sets of documents are released more often as preliminary document versions, and
- people start on the preliminary document versions which increases the risk of re-work and delays

To manage the control complexity it is required to:
- track progress, i.e. status, of small sets of documents or individual documents
- register relationships between components, documents and components and documents to determine the change impact, and
- register who is working on basis of a preliminary version of a document so that they can be notified about the change

4.5 Simulation model of PDP

4.5.1 Brief overview of simulation model

Within the RapidPDM project a simulation model has been developed on the basis of the research model that is presented in this chapter. The simulation model can be used to model a network of document lifecycles that are connected by intradomain and interdomain Start-Finish relationships.
Running the simulation model means creating all the documents according to the pre-defined sequence.
In the simulation model all the steps in the document lifecycle are regarded as jobs that have to be carried out by an engineer with a specific skill. However, not all engineers are entitled to create, review, or release particular documents. This can result in waiting queues if all the engineers that are entitled to carry out a specific job are occupied at a particular moment.

Basically, the simulation model consists of three modules, these are required to control the jobs that are handled by engineers. It involves job release control, job planning and change request handling.

- **Job release control** releases new jobs, which is an action that leads to a change in status of a document. If some jobs in the network are completed or partially completed other jobs can start. Job release control releases this new job to engineers that have the required skills.
- **Job planning** performs different planning activities. An engineer is assigned to every new released job. Data is kept about the pending jobs, which are the jobs in progress, and the jobs that are still waiting to be executed, which are the jobs in the 'in-tray' of the engineers. After assigning an engineer to a job, job planning checks whether this job has a higher priority then the pending jobs of the engineer. If it does, the new job is sent to the engineer and the engineer starts working on the new job. The pending jobs will be delayed because the new job has priority. If the new job does not have a higher priority the job is stored in the 'in-tray' of the engineer. When a job is finished the next job of the engineer will be determined by conforming to the priority and first in first out rules. The progress is reported to job release control.
- **The change request handler** processes change requests. It involves the processing of requests for changing documents. When the change request handler receives a change request it will determine which documents are affected by the change. For the documents that need to be changed a change order will be generated that is sent to Job Planning. There the change order is handled as a job with a certain priority.

For more details on the simulation model is referred to Coolen (2000).

4.5.2 Simulation results

The simulation has been used to study the difference between a sequential and a concurrent product development process. The model showed that under some conditions concurrent product development processes are completed earlier. Important aspects that determine if the concurrent process is faster than the concurrent product development process are:

- degree of concurrency
- propagation of changes
- effectiveness of the review process

The degree of concurrency refers to the degree of overlap between document lifecycles. In the sequential product development process the document lifecycles are executed after one another. In the concurrent product development process the document lifecycles are overlapped. The most extreme situation is where one starts working on all documents at the same time. As one might expect this results in a lot of re-work. Hence it takes more man-hours to complete a project and it also takes longer to complete the project. Somewhere in between there is an optimum where the concurrent product development process is significantly faster than the sequential product development process.

But also the propagation of changes plays a role. It is a parameter that can be set and determines the effect of changes. In the best case scenario only one document is affected. In a worse scenario documents preceding the affected document in the first degree are also affected. Finally, in the worst scenario documents preceding in the second or third degree are also affected document. The more documents that are affected by a change the more re-work and delay of the project is the result. Propagation of changes can be limited by a modular architecture of the product. A modular architecture of the product is essential when starting on preliminary information in order to control the propagation of errors.

Finally, the effectiveness of the review process is important. If the quality of the review process is low, errors are not detected during the review process. These cause changes in later phases of the project, which results in more re-work and a delay of the project. Hence if the review process is poor the effect of concurrent product development is limited.

4.6 Validation of the research model

Although the case studies are presented in section 6 part of it is already presented here. The reason is that one of the case studies, the one at Aero, provided empirical data that supports the research model. In this section a model is presented that is created by one of the employees of Aero. After explaining this model an explanation is provided of how this model matches with the research model.

Aero specialises in four manufacturing processes: sheetmetal production, (high-speed) milling, composites, and assembly. Each of these manufacturing processes consist of a number of ‘standard’ engineering and production steps. Early in a project they make a 'sketch' of the breakdown of the product and for each element in the breakdown they determine what manufacturing process is required to create that element. This is illustrated using the example of a milled part. The standard engineering and production steps for a milled part are as follows:

Engineering	Production Engineering	Production
(C) determine product concept	(P) determine production concept	(F) production & testing
(U) detail 3D model	(G) develop tooling	
(D) capture product definition	(B) develop control system	
	(M) purchase material	

Although some of these steps are executed concurrently they are presented as if they are executed sequentially to keep it simple. For a milled part the following manufacturing processes are required (in sequential order): C, U, D, P, G, B, M, F.

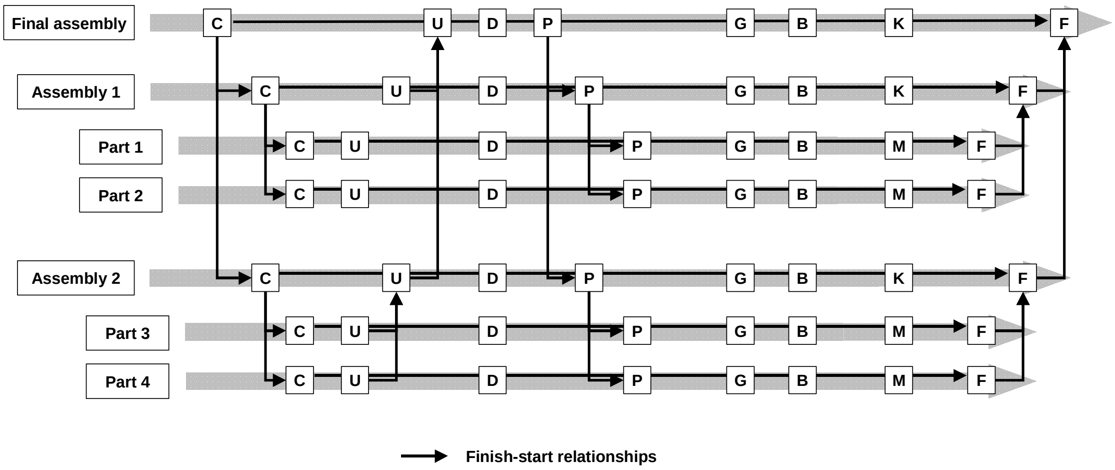


Figure 38 - Concurrent product development at Aero

If you determine the manufacturing processes for each element of the product breakdown it results in Figure 38. At the left side of the figure the breakdown of the product is shown. The large grey arrow at the right side of each element contains the standard engineering and production steps. Part 1, 2, 3, and 4 are milled parts and they do have a predefined sequence of manufacturing steps. The assemblies, assembly 1 and 2 and the final assembly, do have a slightly different chain of manufacturing processes. Instead of process Purchase Material (M) they have process Purchase Items/Joints (K)

The black lines between the individual processes of the *same part* represent a start-finish relationship between these processes. For example, process U of the final assembly should take place after the completion of process C of the final assembly. There are also black arrows between the individual processes of *different parts* that represent a start-finish relationship between these processes. For example, process C of assembly 1 should take place after the completion of process C of the final assembly.

Figure 38 shows a significant resemblance with the research model. Firstly, it is possible to distinguish different domains in Figure 38, *i.e.* a Physical domain and a Process domain. The Physical domain includes the engineering activities C, U, and D while the Process domain includes the production activities P, G, B, M, and F.
Secondly, each domain has its own product model and the processes/activities that are carried out in that domain are directly related to an element of the product breakdown[16]. In my research model, however, there is only one activity related to an element of the product breakdown. But it involves a deliberate simplification to keep the model simple. In Aero's model, however, there are more activities related to an element of the product breakdown. For example, activity C, U and D are related to Part 1.
Thirdly, the decomposition relationships in the product breakdown are also an indicator of the finish-start relationships for the activities in a domain. For example, activity C of Part 1 and Part 2 cannot start before activity C of Assembly 1 is finished. This is similar to the intradomain relationship that was presented in section 4.3.4.4. Finally, Aero also has interdomain relationships as presented in section 4.3.4.4; however, these are not shown in Figure 38 in order to keep it simple. An example is that activity P of Assembly 1 can start early if activity C of Assembly 1 is completed. In other words there is no need to wait for the completion of activity U and D of Assembly 1.

Conclusion
The product development process at Aero can be modelled using the terms introduced in my research model, *i.e.* domains, product breakdowns, intradomain relationships, and interdomain relationships. Because of the many similarities between my research model and the model that is used at Aero there is reason to believe that the research model is suitable for modelling the sequence of activities in the product development process.

[16] Note that the product breakdowns are identical in this case and remember that we used the same simplification in our research model.

5 PDM to control complexity in CE process

This chapter presents the theory about the enabling effect of Product Data Management with respect to starting on preliminary information. It starts with a section on the enabling capabilities of ICT in general to define the term enabler and is followed by a section that describes the enabling capabilities of PDM. In the final section a number of hypotheses are presented that predict how companies start on preliminary information and how they should manage the increased complexity using Product Data Management functionality.

5.1 Enabling capability of ICT

The linking pin between PDM functionality and starting on preliminary information, as presented in the previous chapters, is literature on Business Process Design (BPR). This literature refers to ICT (in general) as an enabler for BPR (Hammer, 1990; Davenport & Short 1990). Therefore this literature is reviewed to come to a definition of the term enabler for this research. Davenport and Short (1990) state that: “*Information technology and BPR have a recursive relationship … Thinking about information technology should be in terms of how it supports new or redesigned business processes, rather than business functions or other organisation entities. And business processes and process improvements should be considered in terms of the capabilities information technology can provide.*”. This recursive relationship is also referred to as ICT being an enabler for BPR.

Davenport and Short (1990) define BPR as: “*The analysis and design of workflows and processes within and between organisations*”. The general trend that they spot is: “*Taylor could focus on workplace rationalisation and individual task efficiency because he confronted a largely stable business environment … Today, responsibility for an outcome is more often spread over a group, rather than assigned to an individual as in the past. Companies increasingly find it necessary to develop more flexible, team-oriented, co-ordinative, and communication based work capability. In short, rather than maximising the performance of particular individual or business functions, companies must maximise interdependent activities within and across the organisation.* ”. This matches perfectly with theory on Concurrent Engineering, while the aim of CE is to create multi-disciplinary teams to improve co-ordination and communication in the product development process. According to the definition of Davenport and Short, the transition from a sequential product development process to a concurrent product development process can be classified as BPR. Therefore it is of interest to know how they define the term enabler. This can be of help to define the enabling role of PDM for this research.

According Davenport and Short (1990) ICT is considered to be an enabler if the application of ICT is used to create new or re-designed business processes that result in huge business process performance improvements, in terms of time, cost and quality. They refer to improvements in performance of a magnitude of 20 percent and higher. They compare these to the relative small improvements that can be achieved by automating the existing process, *i.e.* doing the same but more efficient. It involves improvements in performance of a magnitude of 5-10 percent. The enabling effect is more specifically described as: “*In the broadest sense, all of ICT's capabilities involve improving co-ordination and information access across the organisation, thereby allowing more effective management of task interdependence.*”. This is further explained by defining 8 IT capabilities and their organisational impact, but they also indicate that this list is far from complete and that organisation may want to develop their own list of capabilities that are specific to the processes they employ.

Reviewing BPR literature did not provide a clear definition of the term enabler. It provides a general description of the enabling capabilities of ICT with respect to new or re-designed processes. But at least one thing is clear, they do not mention that the new or re-designed business process is impossible without the application of ICT. This is further illustrated using an example from Davenport and Short. Ford uses IT to make it possible for employees scattered around the world (co-located) to work as a team. The question is whether this co-operation was not possible without the use of IT?
To explain this I use the following situation as a starting point, a co-located team that does not have the following ICT solutions at their disposal: no telephone, no e-mail, and no application sharing. In such a situation it is very hard to co-operate, it is only possible by sending drawings and discuss problems in a letter that accompany the drawings (the option of meeting each other is excluded because they have to work as a co-located team). However, this process is very slow and very inefficient. Introducing telephone will improve the situation, now it becomes possible to send a drawing and discuss the drawing over the telephone. This will improve the speed of the process as well as the efficiency. By introducing e-mail the process is further improved, this makes it possible to send a drawing in less than a minute and directly discuss it over the telephone. When the reviewer has remarks the author can process the changes and send a new version of the drawing to the reviewer by e-mail. In the final step application sharing is introduced, this makes it possible to look at the drawing over the Internet. The reviewer of the model can suggest improvements by indicating them directly on the drawing. The author can see the mark-ups of the reviewer and can immediately process them while the reviewer is watching. This results in a very fast and efficient process with very short iteration loops.

Davenport and Short use the example of co-located teams to illustrate the enabling role of ICT. My explanation shows that it is not impossible to work as a co-located team without the application of ICT. Therefore one cannot conclude that ICT is necessary in the sense that it is not possible without ICT. One can only conclude that working without ICT is not as fast and efficient as working with ICT. This subscribes the statement of Davenport and Short that ICT as enabler for new or re-designed business processes results in an improved business performance.
But this still not leads to a definition of the term an enabler; it only excludes one possible explanation. Using the definition of Davenport and Short, I finally came to the following definition of the enabling role of ICT with respect to process re-design:

"*ICT is an enabler when it allows more effective management of task interdependence*".

In this definition 'more effective' should be interpreted as a better business performance in terms of time, cost and quality and 'management of task interdependence' as better co-ordination and communication between interdependent activities. This definition is used as a starting point to define the enabling role of PDM with respect to starting on preliminary information in the next section.

5.2 Enabling capabilities of PDM

In this chapter the enabling capabilities of PDM functions are discussed with respect to Concurrent Engineering, *i.e.* starting on preliminary information. In section 4.4 it was concluded that starting on preliminary information results in an increased control complexity of the product development process. The increased control complexity is a result of the extra co-ordination and communication that is required for effective management of the (inter)dependent activities. The extra co-ordination and communication is translated in this research as the extra work that is required to:

1. track progress, *i.e.* status, of small sets of documents or individual documents,
2. register relationships between components, documents and components and documents, and
3. register who is working on basis of a preliminary version of a document so that they can be notified in case of a change.

As defined in the previous section ICT is an enabler when it allows more effective management of task interdependence. In this case it means that Product Data Management is an enabler if it supports the above mentioned co-ordination and communication requirements. Each of the three requirements is discussed separately.

Requirement 1

Starting on preliminary information means that it is possible to predict when which type of (preliminary) information should be released. When starting on preliminary information in a structured way it is required to plan the release of preliminary versions of a document.

It starts with planning a project by defining a Work Breakdown Structure (WBS), which involves a decomposition of the project in smaller activities and tasks. The activities result in documents that describe or define the product. In other words there should be a relation between documents and tasks and activities of the WBS.

Besides activity/task to document relationships there can also be document to document relationships. The deliverable of an activity can consist of several separate documents. Therefore it should be possible to define compound documents.

Each individual document has its own lifecycle that is marked by several milestones or statuses. The total lifecycle of a document can be supported by workflow functionality. The events in the workflow will automatically result in a change of a document's status. This change in status is an indication of progress. If progress of each individual document is registered it is possible to aggregate this and to track progress on an activity or project level.

Workflow functionality only makes sense if all documents are available electronically. Moreover, the complete workflow should be processed electronically, including the review process. This requires the availability of mark-up and redlining software.

Summarising the following PDM functions are needed as enabler:

Table 4- Enabling capabilities of PDM for requirement 1

Document Management	Product Management	Process/Project Management
vault management	none	process identification
document identification and numbering		work breakdown structure modelling
document viewing and mark-up		
scanning and imaging		
document structure modelling		work planning
document version control		work tracking
document status & release control		workflow management
document baselines		

Requirement 2

In the case of a change it is important to determine which documents are affected by the change, *i.e.* what needs to be updated because of the change. As mentioned in the previous chapter a change can affect a part (or an assembly) and the documents that are related to the part. Changes can propagate because parts have relationships with other parts and documents have relationships with other documents. Due to the relationships a change in one part or document can also affect another part or document.

To determine the scope of a change it is therefore important to have an overview of all these relationships. If parts and documents are registered in the PDM system it is possible to model these part-to-part, doc-to-doc, and part-to-doc relationships. By following the relationships in the PDM system the user can study the possible impact of a change. Moreover, in a PDM system all affected parts and documents can be put on hold. This means that everybody can see that a part or document is in the middle of a change process. It is also possible to notify the organisation, for instance by e-mail, about the intended change.

In case there are a lot of changes processed simultaneously it is important for everybody working on the project to have an overview of all changes so that they can easily check progress and status of the changes. This is only possible if the engineering change process is handled electronically using workflow management functionality. Reporting functionality is required to provide an overview of the status of the individual workflows of engineering changes.

Summarising the following PDM functions are needed as an enabler:

Table 5 - Enabling capabilities of PDM for requirement 2

Document Management	Product Management	Process/Project Management
vault management	product identification	Workflow management
document identification and numbering	product structure modelling	
document structure modelling	product structure viewing	
document structure viewing	product change impact analysis	
document change impact analysis	product version control	
document version control		

Requirement 3

Because document versions are released preliminary there is a risk that a preliminary document version changes. In this case it is important to know who received a copy of the preliminary document version so that it is clear who should be notified about the change.

Firstly, it is required that each document version is registered and stored in the vault of the PDM system. Then it is possible to log all the actions that are executed on that document version. Secondly a register is required that shows who received a (preliminary) document version. Document versions are distributed if they reach a specific status, such as For Approval or Released for Prototyping. Employees receive a document because they are on the distribution list of that document or document type. Some employees receive a version of the document because they need it as input, *e.g.* an engineer; others receive a version of the document to authorise it, *e.g.* a manager. If the distribution list is stored it is always known who received a copy of which version of the document.

Employees also request a version of a document because they feel they need it. They can request a document version by subscribing to the distribution list of a document or document type. But they can also request a version of the document by searching the PDM system. If the document is found they can view and/or print it the document version they need. A PDM system can log view and print actions, which makes it possible to notify all persons that viewed and/or plotted a version of a document.

Summarising the following PDM functions are needed as enabler:

Table 6 - Enabling capabilities of PDM for requirement 3

Document Management	Product Management	Process/Project Management
vault management	none	(workflow management)
document identification and numbering		
document version control		
document status & release control		

5.3 Performance improvements as a result of PDM

The previous section explained the enabling capabilities of PDM functionality with respect to starting on preliminary information. But the definition also mentions that the enabling capability of ICT, *i.e.* PDM, should result in performance improvements. In this section the performance improvements, *i.e.* the benefits, of applying PDM functionality are described. The benefits are discussed using the requirements mentioned in the previous section .

Requirement 1

Instead of managing one large batch of documents that has to be available at some point in time smaller batches of documents have to be managed that have to be available at different points in time. Therefore multiple due dates have to be monitored at the same time to ensure that the right document is delivered at the right moment. This is achieved by monitoring the lifecycle of each set of documents and each change in document status should be registered. Without a PDM system this data has to be gathered manually, for example by weekly sending a list with all documents or by changing the status in a spreadsheet that is available on the network.

With a PDM system this data is gathered more or less automatically because status changes are registered in the PDM system. Because each change is registered immediately the PDM will always give the latest status about the progress of a project. If status information is collected on paper it is only collected every week or month. In this case it is possible to see the actual status of a project. Moreover, it is also possible to provide an automatic warning if a document is behind schedule.

Every release step the document is routed or distributed throughout the organisation. Due to the preliminary release of documents, several versions of a document are routed through the organisation. With a PDM system the routing or distribution can be done electronically instead of manually which saves a lot of effort in copying and distributing the document versions.

Summary of benefits

- Less administrative effort to track the lifecycle of documents
- Less administrative effort to release (i.e. copy and distribute) a version of a document
- Instant overview of the actual status of documents, activities, and projects

Requirement 2

In order to be able to conduct an impact analysis it is required to register all the relationships between components and parts. In a PDM system it is relatively easy to model the relationships when it is compared with modelling all relations on paper. Moreover, in a PDM system it is possible to browse through the structure in order to do a change impact analysis.

So the benefits of a PDM system are that it takes less administrative effort to register all relationships although it still requires manpower because the relationships are made manually. But the main benefit is that it is easier to determine the impact because it is possible to browse through the structures.

The result is that the impact is determined quicker and that documents and parts are not overlooked easily. The effect of a quicker impact analysis is less re-work because employees can be notified earlier. The effect of not overlooking anything is also less re-work because if

the change is discovered later the amount of re-work is probably higher then when it is discovered right away.
A PDM system can also support the Engineering Change process with workflow functionality. It results in less administrative effort to copy and route the Engineering Change dossier through the organisation. Moreover, the status of the workflow will give information about the progress of the Engineering Change. Without a PDM system this information should be registered manually.
The PDM system can provide every user with an actual status overview of all changes that are being processed at a particular moment. Moreover, if a new Change Request is initiated the organisation can be notified by means of an e-mail.
A PDM system will also prevent the use of a document that is in the middle of a change process that results in the creation in a new version of that document.

Summary of benefits

- Faster impact analysis
- Better quality of impact analysis, not overlooking things
- Less administrative effort to route/distribute and monitor the change process
- Instant overview of all pending changes and their status

Requirement 3
In order to know who is working on a preliminary version of a document it is required to store the distribution lists of documents. As long as there are standard distribution lists this is rather easy. However, if distribution lists are adapted it is important to register and store the changes to the standard distribution list.
In a PDM system the distribution lists of individual documents or document types can be maintained including changes to the distribution lists for specific documents. If the distribution lists are maintained in the PDM system they are always available. Employees do not have to search for the right distribution list because it is stored in the PDM system together with the corresponding document. This means that employees can be notified more quickly, which results in less re-work.
Often standard document distribution lists are used, which means that you receive all documents of a certain type. Without a PDM system it would be too much administration to have a distribution list for each individual document. However, with a PDM system users can (un)subscribe to the distribution list of each individual document. In this case the PDM system does all the administration. Moreover, employees are only notified about changes in documents of their interest.

Summary of benefits

- Less administrative effort for maintaining distribution lists
- Faster notification and hence less re-work
- Possibility to have a customised distribution list for every individual document or document version

5.4 Hypotheses

In the previous sections a theory on the enabling effect of PDM with respect to starting on preliminary information is developed. The theory is based on the theory from Eppinger *et al.* (1997) and Hanssen (2000), but in this thesis I made the translation *from starting on preliminary information* to *starting on preliminary document versions*.

This resulted in the following hypotheses for this research:

Hypothesis 1:
Starting on preliminary document versions means:

- releasing smaller sets of documents,
- which are released more often (in preliminary form) before the final release, and
- employees start on preliminary versions of these documents.

Hypothesis 2:
Starting on preliminary document versions increases the control complexity of the product development process and is managed by:

- tracking progress, i.e. status, of small sets of documents or individual documents
- registering relationships between components, documents and components and documents to determine the change impact, and
- registering who is working on basis of a preliminary version of a document so that they can be notified in case of a change

Hypothesis 3:
The increased complexity caused by starting on preliminary document versions requires the support of PDM functions as described in section 5.2.

Hypothesis 4:
Application of PDM to manage the increased complexity caused by starting on preliminary information leads to the performance improvements as described in section 5.3.

The 4 hypotheses predict the behaviour of companies that start on preliminary document versions and the enabling effect of PDM when it is used to support starting on preliminary document versions. In the next chapter two case studies are described that have been used to collect empirical data to support the hypotheses. This should finally results in the validation of the theory on the enabling effect of PDM with respect to starting on preliminary document versions.

5.5 Conditions for the enabling capability of PDM

In the previous sections is assumed that starting on preliminary document versions increases the control complexity. Moreover, it is assumed that this increased complexity always requires the support of PDM functionality. However, it is possible to think of situations where this is not true, it concerns companies where the complexity of the product development process is low. If the complexity is low the application of PDM as an enabler will not result in impressive performance improvements. Therefore it does not make sense to apply PDM in these situations.
In my opinion there are at least four factors that determine the complexity of the product development process. The factors are discussed in the following four sections. In the next

chapter the complexity factors are used in the selection process of the case study companies. Companies with a low complexity are not suitable for this research.

5.5.1 Complexity of a product

There are several measures of the complexity of a product:

- The <u>number of parts/components</u> of a product. The complexity of a product increases with the number of parts/components.
- The <u>number of functions</u> of a product. Especially in electronics there is a trend towards more functions on a single component. Hence the number of components decreases, but not the complexity because the number of functions increases.
- The <u>number of employees</u> involved in the development of a project. The complexity of a product increases, as more employees are required to develop it.
- The <u>amount of documentation</u> that is required to define and describe the product. The complexity of a product increases if more documentation is required to describe it.

The complexity of a product is lower in the case of a standardised product. Standardisation of the product will stimulate re-use of the modules and parts. This means that these parts and modules do not have to be developed anymore and that the documentation is already available. Although re-use does not results in less parts of a product it results in less employees to develop the product and less documentation to describe the product.

If the product complexity is low it is assumed that the PDM system is not required as an enabler for starting on preliminary information because of the following reasons. If the complexity is low it will be relatively easy to track the progress of each individual version of a document because the number of documents will be low. If a large part of the product is standard the product structure will also be relatively simple. Therefore, it will be relatively easy to determine the impact of a change without the aid of a PDM system. Finally, if the product complexity is low there are few employees involved in the development process. In that case it is easy to notify the right employees of a change because everybody knows from each other what they are working on.

5.5.2 Modularity of a product

Erens (1996) uses the term product architecture for a modular product:

> *"A set of modules connected through interfaces and performing a certain operation. A product architecture partitions the solution space of design, sets conditions for a further decomposition of these modules and specifies the application of these modules in a bigger whole."*

In other words a modular product is decomposed in independent modules and interfaces between these modules are described. Independent means that a change in one module does not result in a change in another module if it does not affect the interface. Therefore a change has only to be communicated with a relatively small group of people, the (multidisciplinary) team that is working on the module.

Moreover, interfaces prevent errors propagating from one module to another. Hence a change will often be limited to one module. Without clear interfaces an error in one module will easily lead to errors in another module.

If the modularity is high it is assumed that PDM is not required as an enabler for starting on preliminary information for the following reasons. If the modularity is high separate teams can work simultaneously on each module. In such a situation each team is responsible for just one part of the whole product. For a team it is relatively easy to track progress because of the

limited team size. This is not true for overall project management who have to track the progress of all different modules, so even in the case of high modularity they will experience complexity.
As an error is often limited to one module it is easier to determine the change impact and to communicate the change to the team members. This is not true if an error crosses the boundary of the interface, resulting in an increased complexity.

5.5.3 Number of changes

The number of changes also affects the control complexity of overlapped product development processes. If the number of changes increases it becomes more difficult to co-ordinate all the changes and to keep an overview. Moreover, if the number of changes increases it becomes more crucial that all changes are processed as quickly as possible.

If the number of changes is high it is assumed that PDM is required as an enabler for starting on preliminary information for the following reasons. It becomes more important to track progress of all the pending changes. Moreover, it becomes more important to determine the impact of a change and to notify everybody as quickly as possible of a change.
If there are only a few changes then the product development process is not disturbed that much, *i.e.* is less complex, and it is easier to handle changes with a paper based system.

5.5.4 Geographical spread

In overlapped product development processes fast communication is very important in order to reduce the impact of changes. Such communication is most effective if the designers and engineers work in co-located teams. This means that members from several disciplines who are working on the same part of the product are co-located for the duration of their involvement in the project. This means that they can overhear each other's conversations and that in case of an 'emergency' they can sit around the table to solve the problem. In such a case you do not need e-mail to communicate with each other and also sign-off workflows do not make much sense.
If members of a team are distributed around the globe it is much more difficult to set up a good and fast communication between the members of the team. Moreover, in case of an emergency it is more difficult to set up a meeting with all the team members. Distributed teams on the other hand require good IT support in order to operate effectively as a co-located team.

If the geographical spread is low it is assumed that PDM is not required as an enabler for starting on preliminary information for the following reasons. If all team members are located in the same office it is easier to track progress then when all members are spread over several locations. The same is true for determining the change impact and notifying everybody about the change, this is easier when everybody is in the same location.

6 Case studies

In this chapter the findings of the two case studies are presented. For both case studies a similar style of presentation is used to make it easier to read the case study section and to make it easier to compare the two case studies. The case study description ends with a section that contains the conclusions.

6.1 Methodology for conducting case studies

In section 1.4 the Research Methodology has been discussed, which resulted in a research design that is shown in Figure 3. This section focuses on the details of the data collection and data analysis phase. Resulting in a more detailed description of the case study design.

6.1.1 Selection of case study companies

The number of case studies is limited to two case studies and should be conducted in companies in the specified research domain (section 1.2). Summarising companies in the research domain have the following characteristics, *i.e.* the dependent variables:

- low sensitivity and slow evolution in the product development process because in this situation it is possible to apply starting on preliminary information according to Eppinger
- pressure to reduce cycle time, otherwise starting on preliminary information is not an issue for the company
- relatively low uncertainty in the product development process so that they can plan the deliverables, *i.e.* documents, that need to be produced
- high product complexity, low modularity, high number of changes, and high geographical spread otherwise the control complexity is too low and it is not an issue to use PDM to support starting on preliminary information

Besides the dependent variables there are also a number of independent variables that are not considered relevant for the selection of the case study companies:

- the industry/market in which the company operates
- the product that is developed by the company
- the production method of the company, *e.g.* mass production or One-of-a-Kind

What is important is that the companies have the characteristics, the dependent variables, which are mentioned above. To stress this two case studies have been selected that operate in different markets/industries, have different products and use different production methods.

For the efficient selection of case study companies he opinion of two experts has been used. The first expert is an expert that wrote a thesis on Concurrent Engineering and starting on preliminary information. The second expert has been working as a consultant, in the field of Product Data Management and Product Development, for more than 10 years.
The companies that were suggested were visited to check to what extent they match with above described criteria and whether they start on preliminary information and use PDM functionality (according the criteria that are described in section 4.4). This finally resulted in two companies that have the best possible match with the described criteria. It is an illusion to think that one can find a company that perfectly matches with all criteria.
The first company is designing and building structures for aeroplanes. In this thesis I refer to this company as Aero. The second company designs and builds complete process plants. In this thesis I refer to this company as Chemical. These two companies comply with the maximum variation criterion because these companies operate in different industries

(respectively aerospace and process), have different products (respectively structural parts of an aeroplane and chemical plants) and use different production methods (respectively series production and one of a kind production).

6.1.2 Criteria for case study design

In section 1.4 was already mentioned that there are several quality criteria that a case study design should meet to guarantee the soundness and reliability of the results. Four criteria have been commonly used to establish the quality of any empirical social research (Yin, 1994):

- Construct validity: establishing correct operational measures for the concepts being studied
- Internal validity: establishing a causal relationship, whereby certain conditions are shown to lead to other conditions, as distinguished from spurious relationships
- External validity: establishing the domain to which a study's findings can be generalised
- Reliability: demonstrating that the operations of a study, such as the data collection procedures, can be repeated with the same results.

These criteria do have implications for the case study design and the way the research is conducted, *i.e.* good working practices. For each criterion there are tactics available to make sure that the criterion is met. In the following the criteria are discussed and applied to this research, resulting in the case study design for this research.

6.1.2.1 Construct validity

Construct validity concerns measuring the right thing, in other words are the defined operational measures the right measures for studying the subject. To increase the construct validity of his research a researcher can use three tactics, namely multiple sources of evidence, chain of evidence, and have the draft case study report reviewed by key informants. These tactics are important during the phase of data collection and in this research all three tactics are used.

First of all multiple sources of evidence are used, such as interviews and documents. The case study at both companies started with exploring how the company starts on preliminary information. During this phase 'evidence' is collected by interviewing employees at project management level and by doing a document study, *i.e.* reviewing Project Plans and Quality Manuals. After the initial exploration the companies were studied in more detail by means of more interviews and a more detailed document study. During this phase project members from different disciplines were interviewed to study the subject from different angles. Moreover, this offered the opportunity to crosscheck the answers of the different interviewees.

Secondly a chain of evidence is available that shows which data has been collected and how conclusions are derived from this data, like in a criminal investigation. In this research the chain of evidence of both case studies is reported in a case study report that is presented in chapter 6. Finally, the case study reports have been reviewed by a selected group of employees in the case study companies to validate the contents of the case study reports. The reason that only a selected group of employees reviewed the findings is because of the conceptual nature of the findings.

6.1.2.2 Internal validity

When a researcher tries to determine whether factor x led to event y, he must exclude the possibility that there is a factor z that actually caused y instead of x. The researcher has to verify whether his inference that factor x leads to event y is correct. This is referred to as internal validity; tactics that can be used to increase internal validity are pattern matching, explanation building, and time series analysis. These tactics are important during the data analysis phase and the first two tactics are applied in this research. The third tactic is meant for studying trends, which means that the researcher infers that several events occur after each other in time. In this research trends are not the subject of study and therefore this tactic is of no use here.

Pattern matching is more useful, which means that an empirical pattern is matched with a predicted (*i.e.* theoretical) one. If the patterns match it can strengthen the internal validity of the case study. In this research the predicted pattern concerns the predictions that have been formulated in section 5.4. The pattern consists of a number of predictions that are related. Together they form a pattern of inferences with respect to starting on preliminary information and the use of PDM functionality. The goal of the case study is to compare the predicted pattern with the outcome of the two case studies that are conducted. If there is a match it strengthens the internal validity of the case study results and hence the theory that is developed.

The other tactic that is used to increases the internal validity is explanation building, which is in fact a special type of pattern matching (Yin, 1994). Explanation building is used to analyse the case study data by building an explanation about the case by stipulating a set of causal links about it. Explanation building involves logical reasoning to derive conclusions, which is something different then explaining a case in a narrative form. Together with the pattern matching strategy, explanation building strengthens the internal validity of the case study results.

6.1.2.3 External validity

External validity concerns the fact whether the results of the case study can be generalised. A common complaint about case studies is that it is difficult to generalise from one case study to another. In survey research selecting a large sample representing the studied population solves this problem. If the same logic were applied to case studies it would mean that a large sample of case studies should be involved.

Yin (1994) stated that a researcher should not fall into the trap of trying to select a 'representative' case or set of cases. Because the set of cases will never be large enough to satisfactorily deal with the complaint that is difficult to generalise the result of case studies. Case studies rely on *analytical generalisation* instead of *statistical generalisation*. The aim of analytical generalisation is to develop a theory and not to statistically proof for instance a relation between two phenomena. Therefore authors such as Yin, Hutjes and Van Buuren, Swanborn, Straus and Corbin, Miles and Huberman argue that only a small number of cases is required to achieve analytical generalisation.

In this research the developed theory is tested in two cases. In section 1.4 was already mentioned that time and resource constraints resulted in the decision to conduct two (identical) case studies, *i.e.* the aim is to get similar results. In section 1.4 was also mentioned that the technique of maximum variation is used to increase the likeliness that the results can be generalised to a broader domain. Principally the results of this research are valid for the research domain that is specified in section 1.2.

6.1.2.4 Reliability

The fourth and final criterion is reliability; another researcher should arrive at the same conclusions when he follows the same steps. The goal of reliability is to minimise errors and biases in the study (Yin, 1994). There are two tactics that can be used to increase the reliability of a case study, namely the use of a case study protocol and the development of a case study database.

A case study protocol involves the preparation of the case study by defining the steps of the case study, select the employees to interview and so on. If somebody wants to check the research he can conduct the same steps in order to see if he arrives at the same conclusions. A case study protocol is especially important in a multiple case study design in order to ensure that the case studies are conducted in a similar fashion

The case study protocol that has been used for this research served as a guideline for conducting the case studies. The term ‘guideline’ is used here because there was no fixed set of questions that was used at both companies. The hypotheses that are formulated in section 5.4 indicate the direction of the interviews and the document studies that are conducted at the case study companies. In addition the following two requirements have been formulated:

- At both companies people from different disciplines and different levels in the organisation should be interviewed
- At both companies similar documents should be reviewed to get a better understanding of the (formally documented) processes

But only a case study protocol is not sufficient to follow the trail of a researcher. The reviewer should also have access to the data that has been collected during the case study. This criterion is met by maintaining a case study database. Therefore a file is maintained that contains case study notes, sketches made during interviews, and company documents (or references to it) such as procedures, project plans and planning schedules.

6.2 *Case 1 – Aero*

6.2.1 Company introduction

The Aero company is part of a larger group that employs approximately 20,000 people. Approximately 3,750 of these people work for the Aerospace Group, which realised a turnover of 335 million Euro in 1999. Aero operates as an individual profit centre of the Aerospace Group and employs approximately 800 people. All activities are centralised in one office except for some milling operations that are still located at another location.
Aero specialises in designing, developing, manufacturing and servicing structural airframe components, assemblies and systems of commercial and military aircrafts and helicopters. Some examples of their products are wheel doors, pressure bulkheads (Figure 39, left), and J-noses (Figure 39, right).

Figure 39 - Examples of Aero's products

In their projects Aero works with commercial and military aircraft integrators and major subcontractors. Aero can handle production to customers' specifications (build-to-print), engineering for manufacture (co-makership), or the complete development and production process working to the customer's functional specifications (main supplier).

The technologies that are applied by Aero concern metal bonding, composites, sheetmetal production, high-speed milling and assembly together with the corresponding engineering. With respect to high speed milling Aero focuses on High Speed Machining of complex parts with a combination of very deep pockets and small corner radii when coping with the ratio between the diameter and the length of the cutting tool is the issue.
Besides high speed milling it is also known for its thermoplastic and thermoset composite technologies. These materials replace traditional materials and the advantage is their low-cost processing and lightweight properties.

6.2.2 Product development process

The case study focused on a project that is typical for Aero, the Pressure Bulkhead project. The details of this project can be found in Appendix 2. In this section the product development process of Aero is discussed in more detail.

6.2.2.1 Overview

In the pressure bulkhead project Aero acts as a main supplier which means that they take care of the complete development and production process. The development process that is used for the development of the pressure bulkhead can be classified as a customer order driven process. In terms of Cooper (1993) it concerns a development process that can be classified as '*additions to existing product lines*' and/or '*improvements to existing products*' (see section 3.1.3).

The development process of the pressure bulkhead project consists of 4 main processes: Engineering, Production, Assembly, and Certification. Activities that precede engineering, *i.e.* the proposal phase, are not discussed here because it is not the focus of this research.
Figure 40 shows (simplified) how the processes are overlapped. Apart from the 4 processes that have been mentioned, the process of Series Production and Sustaining are also shown. It emphasises the difference between the non-recurring process (above the dotted line) and the recurring process (under the dotted line).

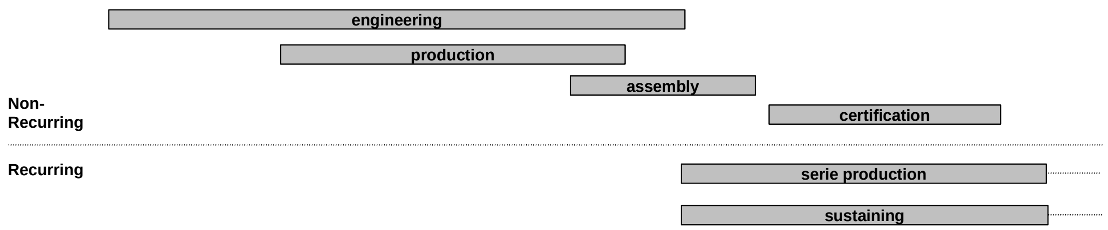


Figure 40 - Overlap of processes at Aero

The non-recurring processes concerning the development activities that finally result in the prototype are executed once. After that the product is taken into a series production, which is called the recurring process. The dotted lines at the end of the series production and sustaining process indicate that these processes can continue for a long time (more than 10 years).

Aero's objective is to have a controllable and therefore a synchronised development process. This is accomplished by the application of phases or stages in the development process, a so-called stage gate process (Cooper, 1993). The three major phases are indicated in Figure 41. In this figure the phases are shown for the non-recurring process.

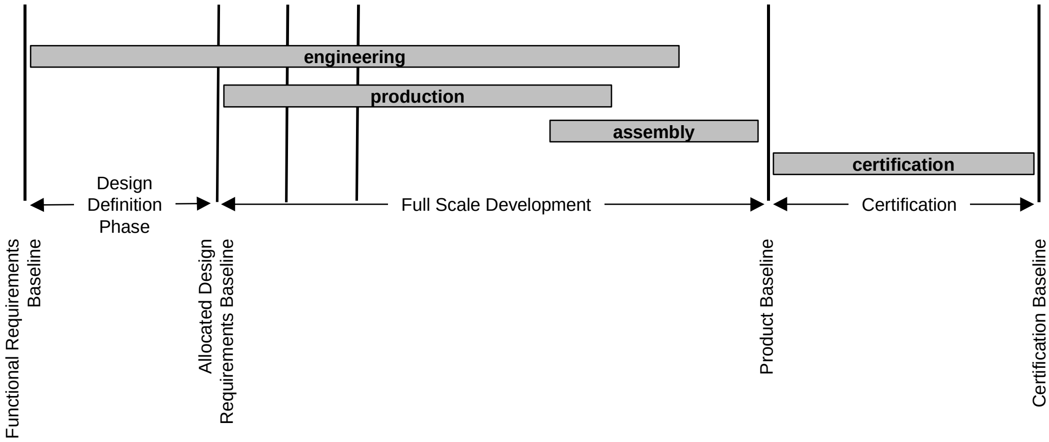


Figure 41 - Major phases in the non-recurring process at Aero

The phases that are distinguished are Design Definition Phase, Full Scale Development and Certification. At the end of each phase they deliver a baseline, which contains the information that serves as a starting point for the next phase. Aero and the customer (the system integrator) review the baselines at the end of each phase[17].

In the following sections the main processes and phases of the development process are discussed in more detail. The focus is on the activities including their inputs and outputs that are of interest from the viewpoint of starting on preliminary information.

6.2.2.2 Main Development Processes

This section provides an overview of the activities that are conducted in each process. The activities are numbered and are for reference purposes in the next section.

Engineering

1. Provide input for finalising the Frontier Models and Work Package specifications by co-operating with the system integrator at their premises. A Frontier Model is a wireframe model that is used to define the interfaces with other parts of the aircraft. Another important model is the Space Allocation Mock-up. This model determines the space in which the product of Aero and other suppliers should fit.
2. Based on the specifications that are provided by the system integrator, Aero makes a conceptual definition of the product. It involves sketches/drawings, called Design Principles, of principal solutions to realise the functions of the product. An example of a function is to provide a load path. Creating the Design Principles also involves conducting a stress analysis to test if the principal solution can bear the loads that have been specified by the system integrator.
3. When the Design Principles are finished they start by detailing the product concept as laid down in the Design Principles. It starts by making 3D models of the mono parts, which are also used to 'generate' a digital mock-up of the product. After that they make

[17] The term baseline stems from Systems Engineering and Configuration Management theory

production drawings (called ‘beeld-blad’) and installation drawings. The production drawing shows the tolerances of each part, which cannot be shown in the 3d model of the part. The installation drawings describe the steps in which the product should be assembled.
Each Design Engineer co-operates with one or more Stress Engineers who check to ensure the mono parts can bear the loads. Their goal is to fulfil design requirements, for instance make the product as light as possible without jeopardising the safety requirements. The outcome of the stress studies of the Design Principles and the Mono Parts is reported in the Preliminary Stress Reports (PSR) and are input to the certification process of the pressure bulkhead.

Production (Engineering)

1. The product is a custom-made product that requires custom-made processes and tooling. This tooling is developed by Production based on information provided by Engineering and the customer. In this case the assembly jig is an important piece of tooling that is required for assembling the product. For the design of the assembly jig Production needs drawings from Engineering showing the rough geometry including tolerances as well as the position of the tooling holes in the pressure bulkhead.
   Other tooling involves moulds for the sheetmetal production. Production needs drawings of the mono parts from Engineering in order to be able to make the moulds. The production of the tooling is outsourced to third parties (*e.g.* low wage countries).
2. Bill-of-Material data is derived from the Design Principles by Engineering and stored in a separate spreadsheet. This spreadsheet is used by Purchasing to order the raw material and parts (for example fasteners). Raw material and fasteners can be a bottleneck because of their lead-time of several months. Therefore, it is important to order these materials and parts as early as possible. Selecting a specific type of fastener early in the process can lead to design requirements because it can influence other design choices.
3. Some parts of the product are produced using milling machines. It involves so-called NC milling machines, which means that they have to be programmed. The solids, *i.e.* 3D models, provided by Engineering are used to generate the NC programs (CAD/CAM integration). Generating the NC programs and milling is performed at another location.
4. Besides milling, sheetmetal production is an important production process. The preparation of sheetmetal production activities involves the routing of the sheets on the shopfloor. Moreover, it also involves planning the number of products that can be cut out of one sheet, *i.e.* nesting.

Assembly

Based on a workorder they start with the sub-assembly of the panels. After the sub-assembly they build the main assembly using the assembly jig. If the assembly is finished the pressure bulkhead is checked. The geometry and tolerances should meet the requirements. After checking it is shipped to the production plant of the system integrator.

Certification

For certification of the product several tests have to be performed on the product. Aero is responsible for tests at component level and the system-integrator is responsible for tests at system level. These tests are witnessed and approved by an aerospace authority. For the tests Aero has to develop custom-made testing equipment.
Besides physical testing this process also involves demonstrating formal compliance with certification requirements by means of the justification dossier (amongst others this includes

the stress analysis reports). The certification process ends when the aerospace authority accepts Aero's compliance demonstration.

6.2.2.3 Phases in development process

In this section the phases of the development process at Aero are described. Moreover, the relation between phases and baselines is discussed.

Start of development process or project

The starting point of the Aero activities is marked by the Functional Requirements baseline that is provided by the customer, *i.e.* the system integrator. The most important part of the baseline is the Functional Work Package Specification. It involves documentation that describes the functional and interface requirements plus the definition of the verifications required demonstrating the achievement of those specified requirements.

Design Definition Phase (DDP)

During this phase the first two engineering activities (mentioned in the previous section) are conducted. It involves finalising the frontier models and the space allocation mock-up. Based on these models a first draft of the design principles is developed. This phase is partly performed in a joint effort, called the Common Design Phase (CDP), at the premises of the system integrator.

During this phase a conceptual model of the product is developed and this phase is characterised by a lot of iterations. It is comparable with other conceptual engineering phases that finally result in a *product architecture description* of the product.

The end of the Design Definition Phase is marked by the Allocated Baseline that is approved during a review meeting. The approved Allocated Baseline contains the Functional Requirements baseline including Design principles, Frontier models, Manufacturing Concept, and additional memo's (containing for instance information about loads).

Full Scale Development (FSD)

In this phase a detailed product model is developed, defined and verified under the responsibility of Aero's Engineering department. It involves activity 3 from the engineering process and all production and assembly activities.

The Product Baseline marks the end of the Full-Scale Development phase and is used as a starting point for the Production and Purchasing activities. The Product Baseline consists of 3D models, production drawings, installation drawings, Bill-of-Materials, and Preliminary Stress reports.

Certification

This phase covers the certification process, as described in the previous section, and starts after the Full-Scale Development phase. The end of the certification phase is marked by the Certification Baseline. This baseline comprises of the Product Definition Dossier and the Justification Dossier of the pressure bulkhead.

6.2.3 Concurrent Engineering in the product development process

This section discusses how Aero achieves Concurrent Engineering in their product development process by conducting activities in parallel, using multidisciplinary teams and starting on preliminary information. Although conducting activities in parallel is not considered to be Concurrent Engineering it is discussed here because it is used to reduce cycle time. Conducting activities in parallel is considered to be a scheduling technique.

Parallel execution of activities

The whole process at Aero is part of a larger process, *i.e.* the development of a complete aircraft. In this higher level process several contractors work on different parts of the aircraft in parallel. This is realised by a good definition of the interfaces that is laid down in the Frontier Models. Such a Frontier Model is an interface document that is put under version control. Aero only needs the interface information in order to develop the pressure bulkhead. They do not need other information from the other contractors. The parts that the other contractors are working on are principally regarded as black boxes. The interface definition between the 'black boxes' should ensure that the parts fit together when assembled by the system integrator.

Aero also executes internal activities in parallel such as detailing the Design Principle (activity 3 as described in the previous section). Parallel execution of activities requires the definition of a product architecture. In this case the Design Principles represent the product architecture. Detailing the Design Principles in parallel is possible because the interfaces between the Design Principles are known. The interfaces are laid down in the Overall Design Principle (*i.e.* the interface document), which shows the complete pressure bulkhead and all the Design Principles that are defined.

Multidisciplinary teams

Aero tells their customers that they are using Concurrent Engineering in their development process.

> *"Aero uses Concurrent Engineering as a systematic approach to the integrated, design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the product life cycle from conception through disposal, including quality, cost, scheduling and user requirements."*

This is called the multi-disciplinary aspect of Concurrent Engineering. At Aero it involves a multidisciplinary team, called the Design Built Team (DBT). Representatives of the different disciplines that participate in the DBT meet weekly. During these meetings the critical parts as well as progress of the project is discussed. Besides the meetings the disciplines co-operate by means of so-called liaisons. Via these liaisons engineering receives early feedback from Production and Purchasing with respect to their design. The DBT team and the liaisons is Aero's solution to developing a design that considers all elements of the product life cycle.

Preliminary information

At Aero they apply case 1 and 3 of Eppinger's Framework (see section 3.3.3.1) to overlap their product development processes. Case 1 means that they do not release all documents in one batch but that they release individual documents when they are ready. This is for example true for the Design Principles and the Product Definitions where every Design Principle and the Product Definition is signed-off and released individually. Therefore, there is no rule that says that all Design Principles should be released at the same time (left side of Figure 42). As soon as an individual Design Principle is checked and approved it is released (right side of Figure 42).

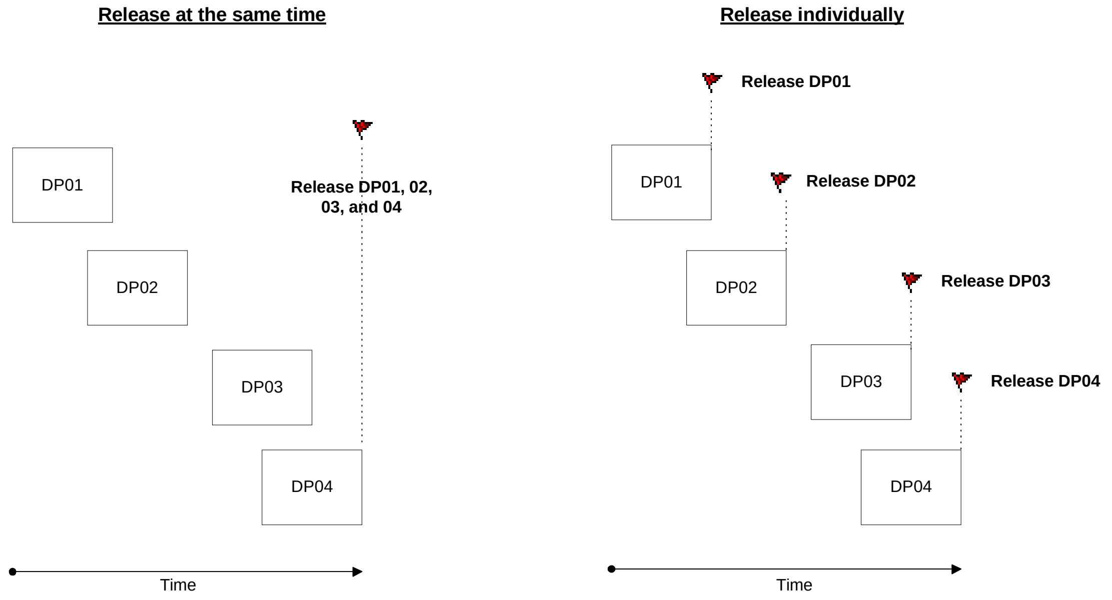


Figure 42 - Two different approaches for releasing Design Principles

But this not interpreted as starting on preliminary information in this thesis. In section 4.3.4.1 starting on preliminary information is defined similarly to case 1 of Eppinger's Framework. Aero starts on preliminary information at three places in their product development. Firstly, Aero starts its Engineering activities on preliminary information from the system integrator, *i.e.* preliminary load loops and frontier models. Secondly, they start on preliminary information on the interface between Engineering and Production. Production starts its activities on preliminary information from Engineering. Finally, Aero also provides preliminary information to their suppliers.

In this case study, however, only starting on preliminary information between engineering and production is studied, because the domain of research is limited to the internal processes only.

How Aero is starting on preliminary information is explained using the research model that is presented in section 4.3. At Aero three domains can be distinguished, it involves a Functional Domain, a Physical Domain and a Process Domain. Each domain has its own product model (see Figure 43). Except for the Functional Domain, as there is no real functional decomposition of the product, there is only a specification that describes the functionality, *i.e.* requirements, of the product. Moreover, there is no Technical Domain because there is a one-on-one relationship between the Functional Solution and the Technical Solution.

The activities in the Functional Domain are executed by the customer in co-operation with Aero (the Common Design Phase) and finally result in the Functional Requirements Baseline. The activities in the Physical Domain start on preliminary information because they start prior to the (final) release of the Functional Requirements Baseline.

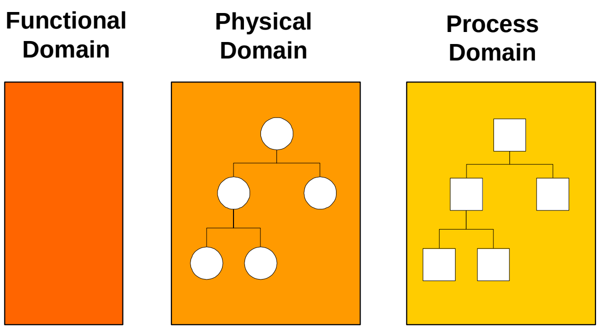


Figure 43 - Domains at Aero

Aero's development process, *i.e.* the Full Scale Development phase, covers the Physical Domain and the Process Domain. Activities in these domains are executed concurrently, *i.e.* overlapped. Activities in the Process Domain start prior to all the activities in the Physical Domain being completed; these two domains overlap each other. On the other hand activities within a domain are executed sequentially, information is only exchanged preliminary in an informal manner.

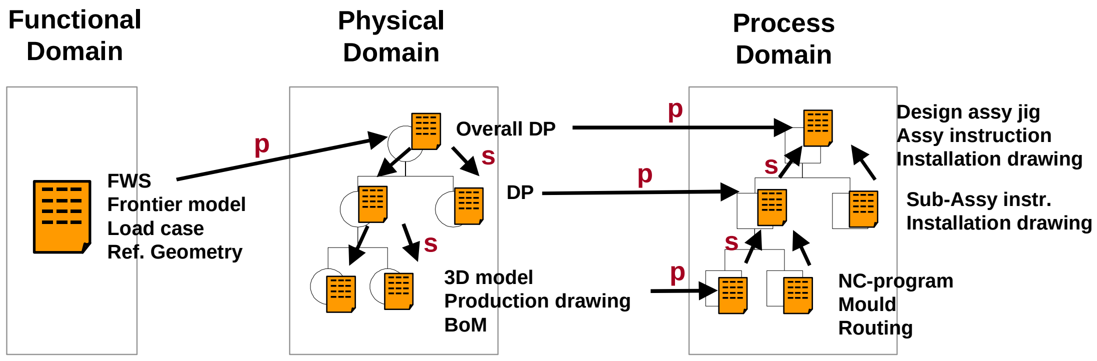


Figure 44 - Relationships between documents in the domains

Figure 44 shows the sequential relationship between the release of documents in the three domains (the arrows). Two types of relationships are shown, preliminary (p) relationships indicating that preliminary document versions are released and sequential (s) relationships indicating that document versions are released sequentially.
The relationships are used to show that activities in the Physical Domain start before all the activities in the Functional Domain are completed because information is preliminarily released. The information within the Physical Domain itself is released sequentially. This means that they first complete the Overall Design Principle, then the individual Design Principles and finally the 3D models, production drawings, installation drawings and BoMs. Experience showed that there are too much iterations in the Design Principles that it is not worth it to start early on the 3D models because it results in re-work.

Finally, activities in the Process Domain start on preliminarily released information from the Physical Domain. They start with the design of the assembly jig based on a preliminarily released version of the Overall Design Principle. The documents within the Process Domain itself are released sequentially.

Summarising Figure A shows that information between domains is preliminarily released while information within a domain is released sequentially (or at least not preliminary released according to a defined procedure).

6.2.4 Design release process

In this section starting on preliminary document versions is discussed in more detail. The focus is on the release of Design Principles and the Product Definition (consisting of 3D models, production drawings, installation drawings, and BoMs), because versions of these documents are preliminarily released. It involves documents that are exchanged between engineering and production.
The preliminary release process is best explained by first describing the 'normal' release process with one final release step and after that the preliminary release process with one or more preliminary release steps before the final release step. In section 6.2.4.1 the 'normal' release process is described and in section 6.2.4.2 the preliminary release process.

6.2.4.1 Release of Design Principle and Product Definition

The release process of the Design Principle and the Product Definition is more or less the same. The first is derived from the latter by omitting certain steps. Figure 45 schematically shows the complete document workflow of the Product Definition at Aero. For modeling the document workflow the same method is used as presented in section 3.2.4.3. In the remainder of this section the steps in the document workflow are described in more detail.

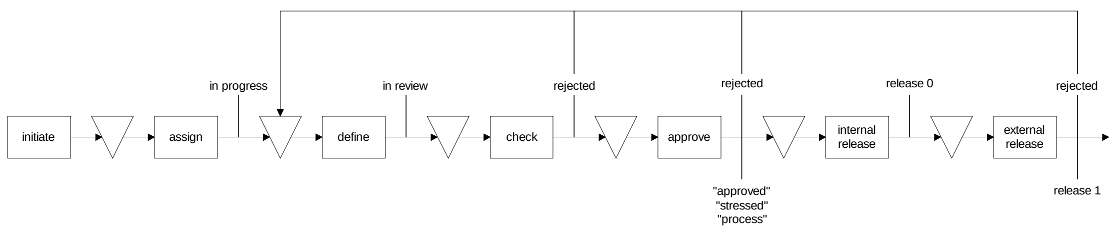


Figure 45 - Document workflow of the Product Definition

Initiate definition of the product (initiate)
The Chief Engineer starts the initiation of the definition of a product. He indicates to the Design Built Team (DBT) leader that a particular Product Definition is required.

Assign drawing number (assign)
Prior to a design engineer being able to start working on the Product Definition the DBT leader consults the Mock-up Integrator for a drawing number. Moreover, they also decide on the position of the Product Definition in the drawing tree. The DBT leader will initiate the CADDS 5 model (3D CAD model) in their PDM system and simultaneously in the PDM system of the customer.

Define product (define)
The design engineer starts working on the CADDS 5 model that has been initiated by the DBT leader. In this phase the status of the Product Definition in the PDM system is IN

PROGRESS. When the design engineer has finished the Product Definition he signs for "compiled" on the routing slip. The routing slip is used to collect the required signatures to release the Product Definition. The Product Definition is routed through the organisation, which means that a routing slip (containing the BoM) and a printout of the production drawing are routed through the organisation. The CADDS 5 model itself is not printed; it is checked-in to the PDM system and receives the status IN REVIEW. The version number of the Product Definition at this moment is A00.

Check Product Definition (check)
An authorised person receives the routing slip (and the production drawings) and looks up the 3D model in the PDM system. He checks the Product Definition for completeness and correctness with respect to the requirements as laid down in the Design Principles. Moreover, a check is made to see whether the model is in line with the guidelines for CADDS 5 models. If no discrepancies are found the person signs for "checked" on the routing slip. The status of the 3D model in the PDM system remains the same, *i.e.* IN REVIEW. If there are any discrepancies the Product Definition is returned to the design engineer who will make the changes. The version number, however, remains the same A00.

Approve Product Definition (approve)
To approve the Product Definition three different approvals are required. First the overall design needs to be approved, then it is approved for stress (*i.e.* strength) and finally it is approved for production. The approval in these 3 steps is realised by the relevant personnel signing the routing slip, with "approved", "stress", and "process". These statuses are not used in the PDM system. If there are any discrepancies found, the Product Definition is returned to the design engineer for corrective action. The status and version of the 3D model in the PDM system does not change after the approval steps, but remains as IN REVIEW and A00 respectively.

Release Product Definition (release 0 & release 1)
When all authorised persons have signed the routing slip the DBT leader releases the Product Definition internally in Aero. In the PDM system the status of the 3D model changes from IN REVIEW to RELEASE 0 and the version number still remains the same, *i.e.* A00.
When the Product Definition is released internally Aero the DBT will request the Mock-up Integrator to transfer the CADDS 5 model to the Vault of the system integrator's PDM system. The status of the 3D model in the Vault of the system integrator then changes from IN PROGRESS to PUBLISH. The DBT leader asks the persons that approved the Product Definition to electronically sign-off the 3D model in the system integrator's PDM system. If all signatures are available the system integrator will check the 3D model. If there are no discrepancies the status of the Product Definition in the system integrator's PDM system changes from PUBLISH to RELEASE 1. In Aero's PDM system the status then changes from RELEASE 0 to RELEASE 1 and the version number still remains A00.
If there are any discrepancies the system integrator will send a message to Aero. These discrepancies will only concern small changes such as a measure or tolerance that is not mentioned in the model. Therefore, correcting the model will not result in a new status or version number of the model. Doing this it is possible to have two identical released versions of a model with the same version number and status indication. After a model is RELEASED 1 it is only possible to change a model after submitting a change request. If the change request is accepted the model is changed and its version number changes to B00.

Release of Design Principles
As mentioned before the release process of the Design Principles is derived from the Product Definition release process (see Figure 46). A Design Principle requires the same checks and approvals but there is no interaction with the Mock-up Integrator nor is it stored in Aero's PDM system or the system integrator's PDM system. To check and approve the Design Principle it is printed and routed through the organisation with a routing slip for approval.

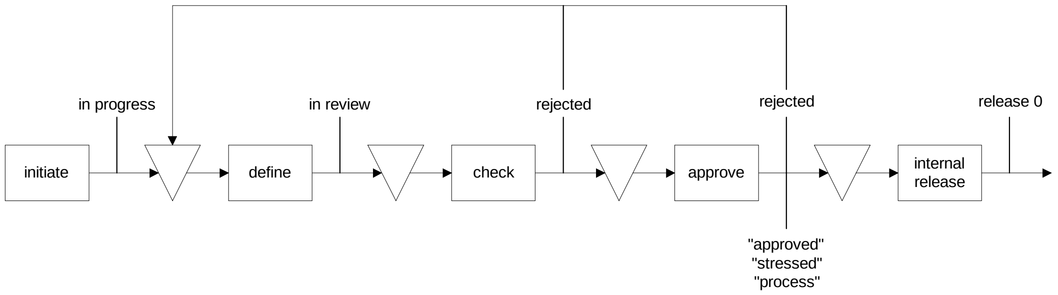


Figure 46 - Document workflow of the Design Principles

The Design Principle is only released internally in Aero and its version numbering mechanism for is the same as that of the Product Definition, *i.e.* A00, B00 and so on. There is no second release step for the Design Principle in which it is transferred to the system integrator's PDM system for approval.

6.2.4.2 Preliminary Release of Design Principle and Product Definition

In this section is shown how the release process, as described in the previous section, is used for the preliminary release of Design Principles and Product Definitions.

Preliminary release of Design Principles
The preliminary release of Design Principles is realised by introducing an additional release step in the release process. Figure 47 schematically shows the release process including the preliminary release step. It shows that they start on preliminary information as (theoretically) expected in and described in section 3.2.4.3.

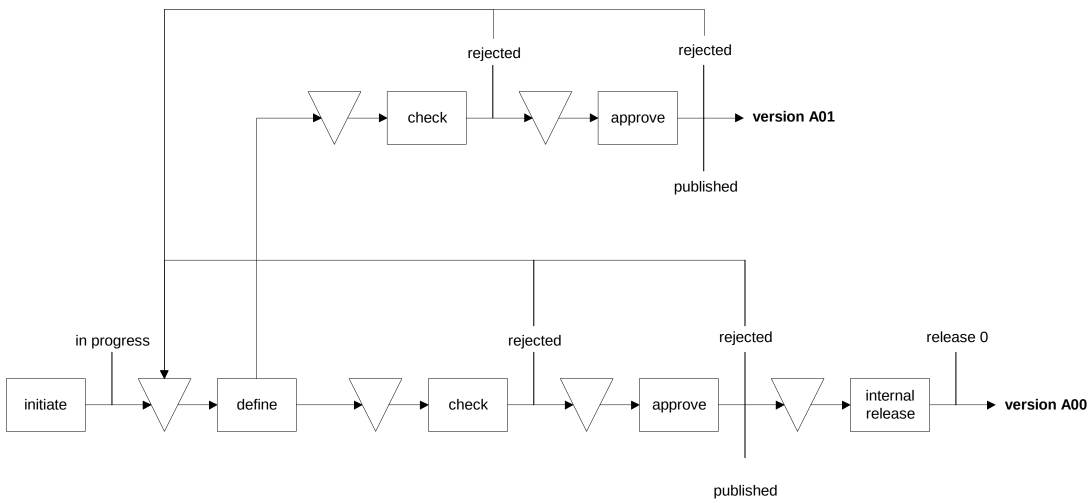


Figure 47 - Document workflow of the preliminary release of Design Principles

Figure 47 shows that it is possible to decide during the Define step to preliminary release a Design Principle. However, before it can be preliminarily released it has to be checked and approved. A preliminary released version of a Design Principle receives the status PUBLISHED and version number A01. Consecutive preliminarily released versions of a Design Principle receive version number A02, A03 and so on. In the final release step the Design Principle receives status RELEASE 0 and version A00.

The preliminary release of Design Principles is not formally documented. However, according to employees from engineering and production several preliminary releases can be recognised that come back in every project. These preliminary releases are as follows:

- In a first release (*e.g.* A01) they derive BoM information from the Design Principles which is put in an Excel spreadsheet and made available to Purchasing. This is used to order raw materials such as sheets and profiles (both are long lead items).
- In a second release (*e.g.* A02) they derive fastener information from the Design Principle. This is a list of fasteners and the quantity of each fastener that should be ordered (fasteners are long lead items too).
- In a third release (*e.g.* A03) the drawings of the Design Principles are provided to production. With this information they can determine the location of the tooling holes that they require for the design of assembly tooling.
- In a fourth release (*e.g.* A04) the drawings of the Design Principles are once again provided to production. However, now the drawings contain more details and based on this information production can design detail tooling such as drilling moulds and (Dutch:) 'stelstukken'.

However, these preliminary releases are not always planned at the start of a project. Often it is determined during the project if preliminarily versions of a document need to be released. Production will ask for the release of a preliminary document version in a DBT meeting (*i.e.* preliminary release by means of a pull mechanism). Engineering will then decide if it can already release a preliminary version of one or more Design Principles. Before a preliminary version is released it has to go through all the check and approval steps.
As the described preliminary process is not formalised it is never certain whether version A01 can be used for Purchasing or not, because this is not explicitly expressed in the *status* of the document version as suggested in section 4.3.4.1.

Preliminary release of Product Definition
By releasing preliminary versions of the Product Definition Aero can start production activities as early as possible. However, the preliminary release of the Product Definition is not formalised. The preliminary release of the Product Definition starts with a preliminary version of the 3D model (status PUBLISH, version A01). Based on this preliminary version Production can make a first draft of the NC-program. In the second preliminary release of the Product Definition the 3D model together with the drawing and the BoM is released (status PUBLISH, version A02). Based on this second preliminary release Production can update its draft of the NC-program. Finally the Product Definition receives status RELEASE 0 and version A00 which means that it is formally released internally Aero. Production can now finalise its NC programs.

Tracking progress of individual documents

At Aero documents are released individually, which means that the progress of these document has to be monitored individually. Progress of documents is maintained in a separate spreadsheet on the network. The DBT leader maintains this spreadsheet what means that he has to manually change the statuses of the individual documents. Hence all status changes have to be communicated to the DBT leader so that he can make the changes.

This spreadsheet is loosely coupled to the project planning. The project planning, a combined Engineering and Operations (E&O) planning, is created and maintained in MS Project. It shows when specific documents have to be available to Production in order to meet the schedule. It identifies document types and not individual documents, hence the spreadsheet contains more detail than the E&O planning.

Bottlenecks in releasing preliminary information

Engineers are not enthusiastic about the release of preliminary document versions because it takes time. It not only takes time to distribute a preliminary document version, it also takes time to check and approve a preliminary document version. When the preliminary release process is formalised several versions of document should be released preliminary. This would result in more check and approval work, which results in overloading the reviewers.

Therefore, engineers prefer the informal preliminary release process, which means that they will not preliminary release a version of any document until production asks for a preliminary document version. In that case they do not have to preliminary release a version of every single document but only those documents that are really required by production. This implies that only versions of documents on the critical path of the project are released preliminary.

6.2.5 Change control process

In this section the change control procedure at Aero is discussed. Special attention is paid to doing a change impact analysis and notifying the organisation about changes. This is of interest from the viewpoint of starting on preliminary information as discussed in section 5.4.

Change procedure

After the first formal release (*i.e.* version number A00) Design Principles and Product Definitions are subject to change control. Everybody related to the project can submit a Change Request by filling in the Engineering Change Request (ECR) form. All ECRs are sent to the Configuration Manager who is responsible for co-ordinating the change process and maintaining the ECR dossier. An ECR dossier consists of several documents. Initially there is only an ECR form, but later on the affected documents are also attached to the ECR. The complete ECR dossier is distributed and filed on paper.

When the Configuration Manager receives an ECR he distributes the ECR to several employees in the organisation. These employees are asked to look at the suggested change and to estimate the impact of the change on the project with respect to time, cost and quality. To guarantee quick processing of Engineering Changes they have to send their comments within three days to the Configuration Manager. Within 5 days after the receipt of the comments, the Configuration Manager organises a Change Control Board (CCB)[18] meeting. During this meeting the CCB rejects or approves the submitted change. After that the customer is also asked for approval to process the change. Based on the comments of the

[18] Members of the CCB are the Program Manager (chairman), Configuration Manager (secretary), Chief Engineer, Lead Manufacturer, Cost Schedule Controller, and Contract Manager.

customer the CCB can take a final decision. If the change request is approved the Cost Schedule Controller allocates additional budget to those departments that are involved in processing the change. After that the Configuration Manager informs the CCB members and gives the initiator approval to make the suggested changes.
The Configuration Manager/DBT leader monitors the status of each ECR in an Excel spreadsheet (Definition Dossier Status ECR). Besides registering the ECRs in their own spreadsheet the ECRs also have to be registered in the customer's system. When all activities related to the ECR are completed and all design documents are updated the change request is closed by the Configuration Manager. By that time all project members are formally notified about the change by sending them an updated version of the changed documents.

There is no formal change procedure for changing preliminary released versions of a document, *i.e.* document versions released as A01, A02, A03 and so on. For changes to these document versions engineers ask the DBT leader for approval who gives oral permission to make the suggested changes. Hence there is not a dossier available for these changes containing all the information about the change.

Determining the change impact

The activity to determine the impact of a change is part of the change process. It means that the feasibility and consequences of a change are estimated. An example is changing the material of a particular part. This requires a brainstorming session of what might be affected by this change. The strength, weight or durability might be affected, which is not wanted. If the consequences are known it should be determined which parts and documents are affected.
At Aero they use the CCB meetings to determine the change impact. In these meetings the possible impact of a change is discussed. As it is a multi-disciplinary team all aspects of the change are taken into account. Based on the experience and competencies of the employees involved in this process they are able to determine which parts and documents are possibly affected by the change. They do not have any aids to support this process except for the knowledge of the employees.
There are some exceptions, however, because some employees maintain all the relationships between parts and documents in a document. But there is no overall document containing all relationships between parts and drawings that are used for determining the change impact.

Registration of preliminary released information

From the viewpoint of starting on preliminary information it is important to know who received a preliminary version of a document. If there is a change to the preliminary released document version, these persons should receive a new version of the document or a notification that their version of the document is changed.
The preliminary released information at Aero receives a formal version number such as A01, A02 etc. which means that it is released in a formal way. When information is released formally it is distributed according to a pre-defined distribution list. Therefore, it is always known who got a preliminary version of a document.

6.2.6 Use of PDM at Aero

In this section is discussed how ICT systems containing PDM functionality enabled starting on preliminary document versions.

6.2.6.1 Introduction

Aero is using so-called PDM system, which is a requirement from the customer. It is used to manage a limited number of document types. In fact only the Design Principles, 3D models and drawings are stored in the PDM system. The main function of the PDM system is retrieval management (storing and finding documents) and version control. Excel spreadsheets containing Purchasing information, Design Principles and (Preliminary) Stress Reports are not managed in the PDM system, for example. These documents are managed in a traditional way on a file server or on paper. Early 1999 Aero selected another PDM system to support their product development process. The implementation of this system is already started and it is expected that it will be completed in 2001.

6.2.6.2 PDM and Document Release at Aero

At Aero the PDM system is used to monitor the paper workflow (of critical documents) by registering the status of the documents in the PDM system. They manually change the status of documents in their own PDM system and in the customer's PDM system. The number of statuses that is managed in the PDM system is however limited. The statuses used in the review process, such as "checked" and "stress", are not registered in the PDM system but are only on the (paper) routing slip. In other words during the review process they can not look up in the PDM system who is reviewing a specific document version at that moment.
One of the major omissions is that they do not use workflow functionality to support the release process because that would substantially improve the above described process.

The preliminary release of documents suggests that it is known when which preliminary document version needs to be exchanged. In other words it should be possible to plan the release of preliminary document versions. A PDM system could be used to plan the documents and to track progress (automatically) on these documents. But at Aero the release of preliminary versions of documents is not planned in the PDM system. They also do not register the planned delivery dates of the final version of documents in the PDM system and they do not actively track the progress of documents in the PDM system. They use a spreadsheet to track the progress of the individual documents. The spreadsheet is maintained by the DBT leader, he manually updates the statuses of the individual documents.

6.2.6.3 PDM and Change Control at Aero

The change control process that is described in 6.2.5 is not supported by the PDM system either, the complete change process is managed on paper. The Configuration Manager co-ordinates this process and also archives the ECR dossiers. The status of each ECR is registered in an Excel spreadsheet, it enables the Configuration Manager/DBT leader to monitor progress. Besides registering the ECRs in their own spreadsheet the ECRs also have to be registered in the customers' system.
Determining the impact of a change is also part of the change control process. However, at Aero they do not use PDM functionality to determine the impact of a change. The PDM system contains a product breakdown and documents that are attached to it. However, it does not contain all the information and therefore it is not possible to do a complete impact analysis. Tooling drawings made by Production, for instance, are not stored in the PDM

system. A change impact analysis in the PDM system will therefore not reveal that a tooling drawing is affected by the change. Instead they use the CCB meetings to determine the change impact. In these meetings the possible impact of a change is discussed. Because it is a multi-disciplinary team all aspects of the change are taken into account.

Some of the interviewees indicate that the PDM system should not be used to manage changes in the design process. They do not want formal change control on preliminary released versions of documents using the PDM system, because it delays the change process. It is more effective to discuss these changes in the DBT meetings with all the persons that are involved. After the meeting the affected documents are updated and a new version is send to all employees involved. The distribution list that is required for doing this is also determined during the DBT meeting.
According to the DBT leaders the PDM system should only be used to manage changes to the as-built configuration. They are also obliged to log these changes because of regulations in the aerospace industry.

A good notification process is important in order to be able to update everybody as quickly as possible about a change. It is important to know who received a version of a document that is being changed. But at Aero it is not explicitly registered who receives a preliminary version of a document. They can only tell that a version of a document is preliminarily released because this reason is mentioned on the document. However, they do not have problems in communicating updates of preliminarily released versions of a document because the distribution lists are rather limited. Most of the time it is the Production Specialist that receives the update and he is responsible for sending it to the engineers that are working on the previous (preliminary) version of the document.

6.2.7 Benefits of PDM at Aero

In this section the benefits of PDM functionality at Aero are discussed. It concerns the benefits of the current PDM functionality that is implemented as well as the future extensions for supporting starting on preliminary information. In the latter case it concerns expectations of the benefits that could be achieved and not benefits that are experienced at the present moment.

6.2.7.1 Current benefits

Currently there is no PDM system at Aero that is used by all employees on all projects. The reason for using a PDM system on the studied project is that the customer demands it. Therefore, only the most essential functionality, *i.e.* the functionality that is required to manage the Product Definition file, is implemented so that the customer requirement is fulfilled. Hence the benefits of using a PDM system are also very limited at Aero.

As mentioned in the introduction the PDM system is used for managing 3D models and the assembly of these models, *i.e.* the digital mock-up. These 3D models and drawings are stored in one place, *i.e.* the vault, and put under version control. Therefore, all engineers have access to the latest version of the models and drawings.As information is stored in a structured manner it is easily accessible. However, the disadvantage is that not all types of document are stored in the PDM system. Therefore, they have to maintain the PDM system as well as a paper archive.
Another benefit that is experienced is the electronic release of document versions to the customer. Before PDM they had to send paper or electronic versions of the document to the

customer. Today they make documents available by placing a version of the document in the vault of the customer. Apart from the fact that it is cheaper it is also faster.

6.2.7.2 Expected benefits of PDM for document release

As they are not using the PDM system for starting on preliminary document versions the interviewees was asked what benefits they would expect from using a PDM system to support starting on preliminary information.

When a company switches to the preliminary release of document versions it has to release document versions in several steps instead of one release step. This means more routing of document versions, *i.e.* copying and distributing of document versions in the organisation. The routing of the document versions is a non-value adding activity that can easily be done electronically using a PDM system. Therefore, applying workflow functionality at Aero to support the preliminary release process will eliminate the non-value-added activities of routing document versions through the organisation.
However, at Aero it is also found that a PDM system cannot eliminate all tasks that are related to releasing preliminary document versions. This was attributed to the fact that releasing a document version also means checking and approval before releasing a document version and logging of all changes after the release of the document version.
Despite the fact that a PDM system reduces the non-value added activities it can not reduce the checking and approval procedures and logging of changes to the preliminary released document versions. At Aero they indicate that the extra approval and check procedures do not outweigh the benefits of reducing the non-value added activities.

Another benefit of a PDM system is that it is easy to keep an overview of all documents and to make sure that the delivery dates of the final version of the documents are met. When starting on preliminary document versions, documents are no longer released in one large batch but released individually. As a PDM system contains status information of documents it can provide the required overview. It is possible to instantly generate an overview of all documents including their status (*i.e.* progress) and scheduled delivery dates.
At Aero the status information in the PDM system is not used to generate status reports. They maintain a separate spreadsheet containing more info than is stored in the PDM system. The DBT leader updates the spreadsheet as he changes the status of a document in the PDM system. Although the number of documents is limited, it is a labour intensive process and requires discipline from the DBT leader. Using a PDM system therefore reduces the manual labour that is required to update the spreadsheet and results in an instant overview of the progress.
The status spreadsheet also contains a planned delivery date for the final version of each document. Sorting the spreadsheet gives an overview of the documents that are due in the next week. Using a PDM system it becomes possible to generate reminders for documents that are overdue. Such reminders can be sent to all employees that are involved in the creation process of the document. This reduces the effort that is required to manually monitor document statuses. Summarising it becomes easier to co-ordinate the process of starting on preliminary document versions.

6.2.7.3 Expected benefits of PDM for Change Control

The Configuration Manager was asked what the benefits would be if the ECR process would be supported by a PDM system. According to him it would reduce the administrative tasks that are required to manage the process. Especially the time that it takes to distribute the ECR

dossier to the employees involved. Moreover, it is not longer required to manually update the status of documents because the PDM system automatically does this.
He was also asked if electronic distribution and tracking of the ECR using workflow functionality would increase the speed of processing a change. However, he did not expect that this would be the case. The employees that are involved in the process determine the speed of the ECR process. Firstly, it takes time before they have estimated the impact of the change. Secondly it takes time before the CCB meets each other. However, if there is an urgent change the CCB can meet within hours instead of days. But in this case the priority increases the speed and not the PDM system.

At the moment they do not use any system support at Aero to determine the impact of change. A PDM system could support conducting a change impact. However, the basic issue is to understand the nature of a change, *e.g.* the consequence of changing the material of a part. A PDM system will not support this because it does not contain knowledge about the application of materials (*i.e.* it is not an expert or knowledge system). It will only assist in more accurately determining which documents are affected when the nature of the change is understood.
It was mentioned that if all the documents at Aero are managed in the PDM system and are also linked to the product breakdown that it would probably help in determining the impact of a change at Aero in terms of the documents that are affected. As they are not using any system at Aero today to support this function one could call a PDM system an enabler for this function. On the other hand they expect that this function would require more registration effort, therefore, it might be not as effective as having a CCB or DBT meeting in which the change impact is determined.

Aero was also asked if they would expect that changes are communicated faster within the organisation using a PDM system; the answer was no. The reason is that changes are discussed during CCB or DBT meetings. During these meetings it is determined which documents are affected. Because all disciplines are represented in these meetings all parties are immediately notified. The representatives of each discipline are responsible for notifying their colleagues about the change. Formal notification after the meeting with a message via the PDM system will probably be slower because it takes time before the information is processed after the meeting and sent to everybody.

6.2.8 Conditions for success

An important prerequisite for realising the benefits of a PDM system is having all documents available in digital form. Otherwise it is not possible to manage and monitor the workflow electronically. In the case study it was found that there are some hurdles to take before all documents are available in electronic form. The following two examples underline the problem of making all the documents available in electronic format.

Example 1: Design Principles
The Design Principles are developed during the Design Definition Phase. As mentioned before Aero uses a CAD system to create the Design Principles. This implies that the latest version of the Design Principle is always available electronically and there should be no hurdle to the preliminary release of the Design Principle using the PDM system.
However, this is not the case because sometimes the actual situation is found on paper master copies and not in the PDM system. The reason for this is that engineers discuss the Design Principle with others. During these discussions they make the changes on a paper copy of the

Design Principle. They indicate the changes with a pencil on the paper master copy. Making an electronic master copy would mean that they would have to process the changes after the meeting in the electronic version of the Design Principle. Then they have to do everything twice, on the paper master as well as the electronic master. Due to the fast iterations during the Design Definition Phase this would result in a lot of extra work.

It would be different if they had equipment where it was possible to mark-up Design Principles electronically. However, they indicate that it would require A1 or A2 screens in meeting rooms to mark-up the Design Principles electronically. Moreover, colour printers are needed to make plots that show the changes in colour. For another projects they do have equipment that makes it possible to mark-up document versions electronically on a large screen in a meeting room. However, this requires a rather high investment that is partly paid by the customer. In the studied project the equipment is not available because there is no budget for these kind of tools in the project.

On of the DBT leaders mentioned that he would like to have a so-called project room or war room for the next project. This is a room that is used by the project team for their meetings and discussions. On the wall one can find the actual situation regarding the planning but also regarding the design. Changes can be easily indicated on a version of the document on the wall. If somebody wants to know the latest status he has to go to the project room. If the design has become more stable (*i.e.* fewer iterations) the documents can be managed electronically using a PDM system but till that time a ‘traditional’ project room is preferred.

Example 2: Stress studies

The first example showed that documents are created electronically and later a document version is printed and managed on paper. After some time they create a new electronic version by incorporating the changes on the paper master into the electronic master. But there are also documents that are not electronically available at all.

An example is the studies that are performed by stress engineers. Stress engineers start a stress study on a paper copy of the Design Principle that they receive from the design engineer. This copy, together with the load cases that are provided by the customer, are used as starting point for their calculations (in an Excel spreadsheet). The outcome of their calculations is for instance the thickness of a component. They report their findings on the (paper) copy of the Design Principle and give it back to the design engineer who incorporates the results in the Design Principles.

Because versions of these documents are managed and created on paper it is difficult to manage the configuration of these files. In other words which version of a Design Principle and which version of the load case led to which version of a stress study and in which version of the Design Principle are these results incorporated.

6.2.9 Conclusions

The previous section presented empirical data to support/reject the hypotheses. In this section the observations are summarised resulting in the conclusions of this case study. The conclusions are structured according to the four hypotheses mentioned in section 5.4.

Hypothesis 1

Although Aero does not formally start on preliminary information one can conclude that they start on preliminary information as defined in chapter 5:

- releasing smaller sets of documents have to be monitored and tracked,
- these smaller sets of documents are released more often as preliminary document versions, and
- people start on the preliminary document versions which increases the risk of re-work and delays

However, the procedure for starting on preliminary information is not formalised and documented in a procedure. This means that they have not:

- defined the content of a document version that is released preliminary,
- defined the point in time that a preliminary version of the document version should be available, and
- defined the decisions that can be taken on basis of this preliminary document version.

This is co-ordinated in a rather informal way via the DBT leader using a so-called pull mechanism. This means that production has to ask the DBT leader to preliminarily release versions of specific documents. DBT leaders indicate that it is not desirable to have a standard release procedure (as suggested in my theory) for the preliminary release of versions of a document because it means more work. If it is not necessary to preliminary release a version of a document it should be avoided. Principally only versions of documents on the critical path should be preliminary released.

The fact that Aero uses the same release procedure for the release of preliminary versions of a document as for final versions of a document results in extra work, *i.e.* more check and approval work as well as logging of changes. This suggests that a less rigid approach to releasing preliminary document versions should be used.

Hypothesis 2

At Aero they use a spreadsheet to plan and track the release of preliminary document versions. This spreadsheet is updated by the DBT leader and contains status information about critical documents in the project. The DBT leader manually updates the spreadsheet each time the status of a document changes.

They do not explicitly register who has received a preliminary version of a document. At the beginning of a project the key-persons that should receive all documentation are identified. They are responsible for the further distribution of the documentation in their discipline. Hence it is relatively easy to quickly communicate a change. Moreover, the DBT approach also contributes to a fast co-ordination and communication of preliminary document versions as well as changes to these document versions.

There was no proof that Aero needs PDM functionality to do a change impact analysis. The impact of a change is determined by using their experience and competencies. Therefore it is not a requirement to have PDM functionality in place to support the change impact analysis.

Although they mentioned that it could be useful to use PDM functionality, they could not tell whether it would substantially improve the change process.

The first finding supports hypothesis 2 while the other two findings do not support hypothesis 2.

Hypothesis 3
The following provides an overview of the PDM functionality that enables starting on preliminary document versions at Aero. This is done on basis of the requirements that were formulated in section 5.2.

The following table provides an overview of the required PDM functions that are used at Aero to support requirement 1.

Table 7 - Required PDM functions for requirement 1 at Aero

Required PDM functions	Aero
vault management	+
document identification and numbering	+/-
document viewing and mark-up	-
scanning and imaging	-
document structure modelling	-
document version control	+
document status & release control	+
document baselines	-
process identification	+
work breakdown structure modelling	+
work planning	+
work tracking	-
workflow management	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

The following table provides an overview of the required PDM functions that are used at Aero to support requirement 2.

Table 8 - Required PDM functions for requirement 2 at Aero

Required PDM functions	Aero
vault management	+
document identification and numbering	+/-
document structure modelling	-
document structure viewing	-
document change impact analysis	-
document version control	+
product identification	+/-
product structure modelling	+
product structure viewing	+
product change impact analysis	-
product version control	+
workflow management	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

The following table provides an overview of the required PDM functions that are used at Aero to support requirement 3.

Table 9 - Required PDM functions for requirement 3 at Aero

Required PDM functions	Aero
vault management	+
document identification and numbering	+
document version control	+
document status & release control	+
(workflow management)	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

From these tables can be concluded that only the basic PDM functionalities are in place and the major omission is the support workflow management, which is a requirement to enable starting on preliminary information.

Hypothesis 4
The employees at Aero was also asked which benefits they would expect if more PDM functionality would be used to support the release of preliminary information (or more specific the release and change control process). According to the Aero employees it will result in a more efficient process because it is no longer required to copy and distribute documents manually. Moreover, it is easier to track progress of the release process of each individual document.
On the other hand it was also mentioned that a PDM system does not eliminate the extra check and approval work that is caused by the release of preliminary document versions. However, using a different procedure for the preliminary and final release of documents will probably improve this situation.

The benefits of using the PDM system to support the change control process are not completely in line with what was expected. The case study showed that a PDM system is not necessarily appropriate for conducting a change impact analysis and for the fast notification/communication of changes. Moreover, it is not expected that a PDM system will accelerate the change control process because the speed is more a matter of priority. On the other hand it was indicated that the PDM system helps to reduce the copy and distribution activities of the change control process resulting in a more efficient process.
The most important requirement for realising the benefits is managing the complete release and change control process electronically. This means that all documents should be available electronically. The case study showed that there are several reasons why documents are not always available electronically.

Other findings
Another finding of the case study is that a PDM system seems to be of little use in a dynamic environment that is characterised by a lot of *creative* iterations. Engineers indicated that they prefer to work with paper master copies in such an environment, because it is quite easy and also quite cheap to mark-up a paper master copy. Sharing these master copies should be realised in a project room. Sharing data in a PDM system in such an environment would require frequent updates of the paper master copies and their electronic sources. This suggests that data should be shared in the PDM system as the design becomes more stable and is released.

Summarising it is not desirable under all conditions to use the PDM system for managing the master copies. If all members of design/engineering team work in the same office it is possible to start with paper master copies. As soon as preliminary versions of a document are released, typically after the creative iterations are completed, the paper master copies should be replaced by electronic masters.

6.3 *Case 2 – Chemical*

6.3.1 Company introduction

Chemical is a leading international EPC (engineering, procurement and construction) Contractor with a strong technology base of more than 50 process technologies, providing the Process Industry world-wide with a full range of services. World-wide they employ approximately 6,000 people and in The Netherlands they employ about 1,700 people.

Figure 48 - Examples of Chemical's products

Since 1954 Chemical, The Netherlands has carried out more than 500 major EPC projects throughout the world and has gained an international reputation in conceptual engineering, detail engineering, design procurement and construction of all types of refineries, process plants, offshore platforms and onshore gas and oil handling and production facilities (see also Figure 48).

6.3.1.1 Case study project

The case study focused on one large project that is being carried out by Chemical. This section provides some details about this project. It involves a very large project because it is expected that PDM is especially required in large projects to start on preliminary information.

Project details
The project involves the Engineering, Procurement, and Construction (EPC) of a Styrene Monomer, Propylene Oxide (SMPO) plant. The total plant is composed of a number of smaller plants that can be regarded as a project on their own. These smaller projects can consist of 75.000 man-hours of engineering work each. The total project involves 1.000.000 man-hours in Engineering and Procurement and 12.500.000 man-hours in Construction. The total value of the project is approximately US$ 500 million.

Project organisation
Although the plant is being built in Asia, the Home Office in The Netherlands conducts the requirements engineering and procurement services. The project is carried out by a multi-disciplinary team that is co-located at the Home Office in The Netherlands. Co-location means that members from different disciplines reside at the same location. In this case several

floors of the Home Office in The Netherlands. The team members are not necessarily all Chemical employees. It also involves personnel from other Chemical offices and personnel from third parties, such as representatives of the customer and Construction partners. For the actual construction of the plant in Asia, Chemical hires local contractors. These local contractors in their turn contract a number of sub-contractors in the region.

Project approach

The SMPO plant in Asia is a copy of another SMPO plant that has been built by Chemical for the same customer. Therefore, Chemical tried to follow a so-called 'copy-concept', by re-using as much of the previous design as possible. This was not always possible because of some technical differences, local authority requirements and recent technology developments.
Besides these differences between the two projects there are also organisational differences. In the first project the main business driver was lead-time or schedule (time-to-customer), while for the second project the main business driver was cost.
The first project had to be completed in a time frame that was shorter than ever before. By applying a Concurrent Engineering approach this was accomplished. The consequence of this approach was that there is not enough time left to optimise the design. This meant, for instance, that more piles were used for the foundation of the plant than were strictly necessary. By over-designing the plant they could anticipate for changes later in the project. If they did not over-design the plant such late changes could have had severe consequences such as delays, which would have been unwanted.
For the second project the main business driver was cost. The design of the first plant should be optimised in this project. This resulted for example in fewer piles and less concrete. Although cost is the main driver, time was still important in this project. Hence they also used a concurrent engineering approach but the schedule was not that tight. Therefore, there was time left for optimising the design as well.

6.3.2 Product development process

Chemical carries out EPC projects for customers in the chemical and petro-chemical industry such as Shell, Exxon, Akzo/Nobel, and DSM. The EPC process of Chemical is a product-oriented, engineer-to-order process; it is carried out on the basis of a customer order.

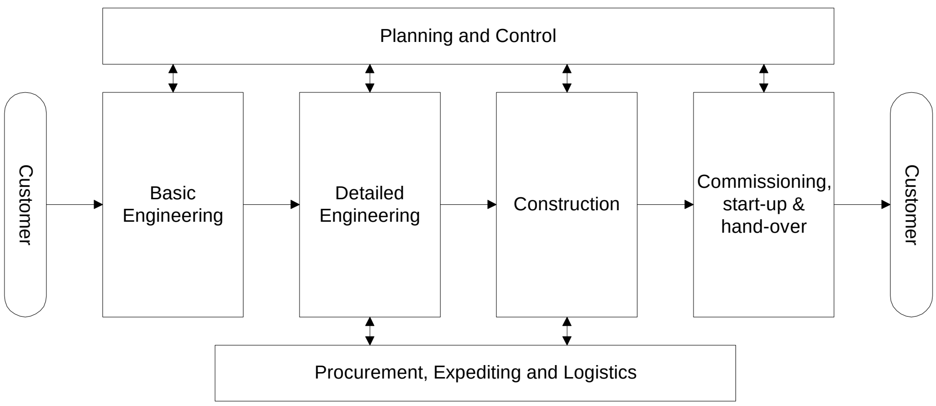


Figure 49 - Main phases of the EPC process

Figure 49 shows the main phases of the EPC process at Chemical. Basic Engineering is not always part of the EPC process at Chemical. The customer or another company can also do the Basic Engineering. For the details of the EPC process please refer to Appendix 3.

6.3.3 Concurrent Engineering in product development process

This section discusses how Chemical achieves Concurrent Engineering in their product development process. It focuses on overlapping of phases and serves as an introduction for the next section on starting on preliminary information.

In the past Engineering, Procurement, and Construction were performed sequentially, also called *Phased EPC*. Hence all data is released in one large batch at the end of a phase. But changing market requirements demanded shorter schedules in the process industry (Krispijn, 2000). This requires a new approach such as *Concurrent EPC* where Engineering, Procurement and Construction are conducted concurrently. In a Concurrent EPC project they start to build the foundation for a plant before the 25% percent point of engineering is reached. While in the case of phased or sequential EPC they did not start with Construction until Engineering was well advanced or nearly completed (*i.e.* 60-100% engineering point).
In the case of Concurrent EPC the Construction and Start-up schedule determine the Engineering and Procurement schedule. Engineering have to deliver the information in the sequence that is required by Construction. Hence in a Concurrent EPC project the data is released in smaller batches as shown in Figure 50. This strategy is similar to *Case 3* of Eppinger's framework.

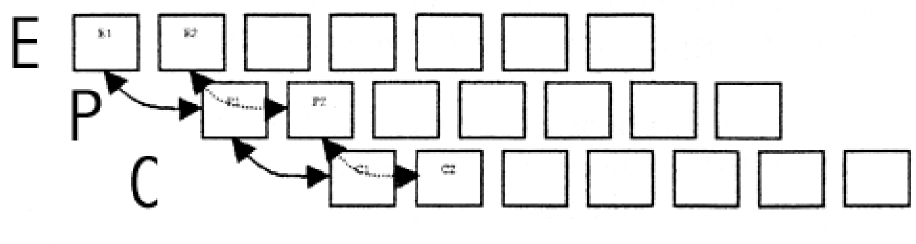


Figure 50 - Concurrent EPC process

This is illustrated using the following example. In Construction they start with the piling of the plant. Hence the piling plan and the specification of the piles is one of the first things that is required by Construction. However, the problem is that in order to develop the piling plan you have to know the weight of the equipment that the piling is supporting. The problem is that you do not have exact figures about the weight and dimension of equipment at the start of the project. This requires preliminary information with respect to the weight of the equipment. This challenge is faced in the majority of their projects.

6.3.4 Design release process

The previous section showed that Chemical changed from a Phased EPC to a Concurrent EPC process. It was already mentioned that this is achieved by releasing smaller batches of information (Case 3 of Eppinger's framework). But they also release preliminary versions of a document (Case 1 of Eppinger's framework). In this section the release process of preliminary document versions, as used by Chemical, is discussed.

6.3.4.1 Preliminary release process (category 1)

Studying the document release process at Chemical revealed that it is possible to distinguish between several categories of preliminary document release. In this section the category 1 document release process is discussed, which is similar to the preliminary document release process that was presented in chapter 4.

The category 1 preliminary release process has pre-defined statuses. Therefore, a document goes through all these statuses (planned). Moreover, the statuses have a pre-defined purpose and a definition is provided to indicate for what purpose a preliminary released document version might be used. Before a document version is preliminary released it is checked for errors (checked). Finally it is registered to whom a preliminary document version is distributed (registered). This is done using the so-called distribution lists that can vary per document type, project and even per document instance. Therefore, this category of preliminary document release can be described as **planned**, **checked** and **registered**.

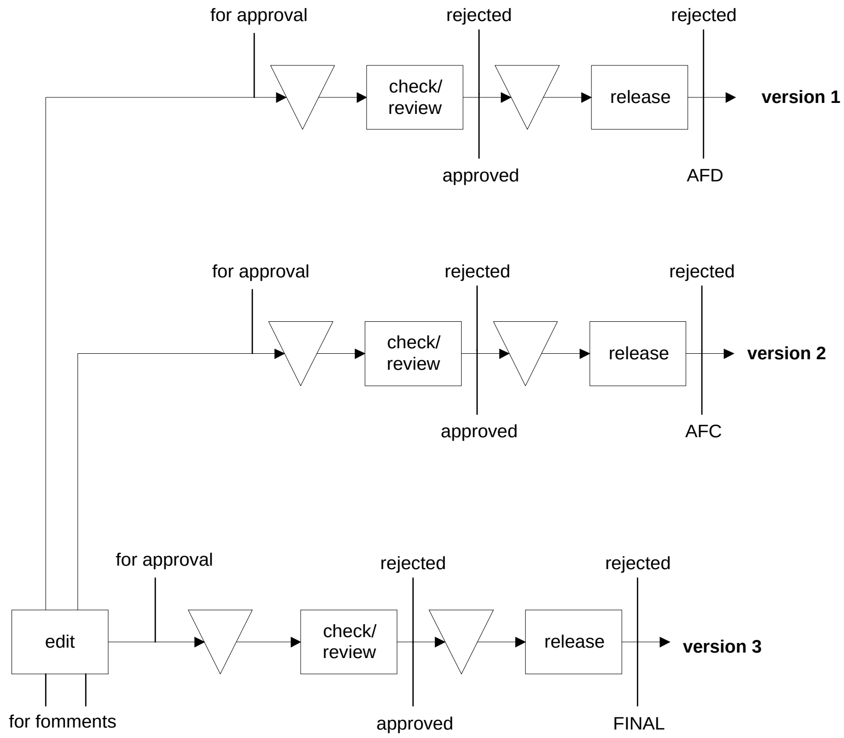


Figure 51 - Document workflow for the release of PEFS

The release process of the PEFS at Chemical is used to illustrate the category 1 preliminary release process, also shown in Figure 51. It shows that they start on preliminary information as (theoretically) expected in and described in section 3.2.4.3.

The lifecycle of the document starts with the creation and editing of the documents; the document is ‘in progress’. If the author needs input from others he can send a versions of the document with status For Comments to a number of colleagues. These colleagues can give their feedback to the author who incorporates the feedback into the design. This feedback is part of the design process and is not a quality assurance or approval step. If the originator thinks that it is necessary he can send several versions of the document For Comments. When the author thinks that the design is ready to be released, a version of the document is sent For Approval. Meaning that colleagues of the same discipline and/or colleagues of other disciplines and/or the customer verify the document version and give their approval. After the approval the document version is released as Approved for Design (AFD). This is the first preliminary release step that is used at Chemical and marks the end of the BDEP phase (see also Appendix 3)

AFD indicates that a document is stable enough to start other design activities that need the document as input. A percentage is used as an indicator for the stability of the document, for AFD this is 60%. The percentage is similar to what is defined as the quality or maturity of the content of a document in this thesis. Once a version of the document is released AFD it is important that all changes to that document version are registered. In the next release these changes should be attached to the document so that others can quickly see what has changed since the previous released document version.

The next preliminary release is Approved for Construction (AFC). AFC marks the end of the EPC phase (see also Appendix 3) and indicates that a document is stable enough to start construction activities related to the document. When a version of the document is released for AFC the probability percentage changes from 60% to 80%. Once a version of the document is released AFC there can still be changes made to the document. It is possible that Construction initiates Field Changes because something does not fit for example. These changes are incorporated in the final release of the document, *i.e.* status As-Built or Final. An As-Built document represents the actual situation on-site.

The preliminary statuses AFD and AFC are used to overlap the Basic Engineering, Detail Engineering and Construction phases. Each phase can be regarded as a domain as specified in chapter 4. The release of document versions is used to sequence the activities in different domains. Therefore, the preliminary release process can be characterised as interdomain relationships.

Note: Appendix 4 contains additional examples of the document release process at Chemical.

6.3.4.2 Preliminary release process (category 2)

Sometimes it is required to use additional statuses such as a status before AFD or a status between AFD and AFC, examples are For Enquiry Only and For 10% Cost Estimate. Both are pre-defined statuses but they are only used when necessary.

A preliminary version of a document can be released as For 10% Cost Estimate before it is released as AFD. This status means that the preliminary document version can only be used for making a 10% cost estimate and not for any other purpose. They will decide to use such an additional status, or unplanned preliminary release, if waiting for the first planned release, for instance AFD, would cause a delay in the project schedule.

Before a preliminary document version is released for 10% Cost Estimate, it is checked by others. Moreover, the author commits itself to the content of the document. This means that he is not allowed to change the content of the document, if that change influences the 10% Cost Estimate, without consulting the user of that preliminary document version. The users of

the preliminary document version are also registered by means of a distribution list. Therefore, in the case of an approved change the employees on the distribution list receive an updated copy of the preliminary document version.
Summarising the release is unplanned, but it is defined that there can be additional preliminary release steps. In other words specific document statuses are reserved to release a version of a document for a specific purpose. Although these preliminary releases are not planned they are checked and registered. Therefore, this category of preliminary document release can be described as **unplanned**, **checked** and **registered**[19].

A special type of preliminary release in this category is the AFC with Holds. This is used to partly release a document version as AFC. When releasing a document version AFC with Holds, the objects that are on hold are excluded from the release. These objects are often marked with a colour or a cloud.

6.3.4.3 Preliminary release process (category 3)

This category is similar to the second category because it involves additional preliminary releases. However, the purpose of these preliminary releases is not pre-defined like the preliminary releases in the first and second category. The statuses that are used for such a preliminary release are called For Information or Preliminary. This status is similar to the status Posted that was mentioned in section 3.2.4.3.
There can be various reasons why somebody already wants to have a preliminary version of a document. In such a case the preliminary document version receives a general status such as Preliminary or For Information. Such a status does not tell anything about the purpose for which the document can be used.
A preliminary document version with status For Information or Preliminary is not checked before it is issued. Moreover, the content of the document version is not frozen. Therefore, the author can change the content of the document version without consulting the receiver or user of the document. Finally the author is not obliged to notify the receiver about the changes, *i.e.* there is no distribution schedule for the preliminary document version.
To summarise this preliminary release is unplanned and there is only a general definition of its purpose, *e.g.* For Information. Often it is not checked before it is released and there is no distribution list. Therefore, this category of preliminary document release can be described as **unplanned**, **unchecked** and **unregistered**.

6.3.5 Change control process

When a version of a document is released, for example AFD, the change control procedure becomes effective. From that moment the document version is frozen and it is not longer allowed to make radical changes in the design because this can cause extra work and delays. The person that wants to make a change has to submit a Change Notice. A Change Notice is a standard form on which the submitter indicates the reason for the change, the number of man-hours and so on. Affected documents are attached to the Change Notice and the requested changes are indicated on version of these documents. Disciplines that are affected by the change are asked to make a 10% cost estimate of the change.
Typically lead engineers of one or more disciplines determine the impact of a change. In the case of a comprehensive change the lead engineers will discuss it in a meeting. Based on their

[19] Unplanned does not refer to schedule issues in this situation, because in a project they plan a For 10% Cost Estimate release. It only refers to the fact that the status it is not always part of the document lifecycle.

knowledge they know which part of a design is influenced by a change and hence which documents need to be adjusted. The Project Manager evaluates the Change Notice, if the value exceeds US$ 50K he determines whether it involves a contract change. In the case of a Contract Change Chemical can decide to ask the customer for additional budget. Otherwise it is an Internal Change or Field Change and Chemical does not receive additional budget.
When the project manager approves the Change Notice the changes should be made as specified in the Change Notice. This finally results in the release of a new version of the affected documents. The employees that are on the distribution list of that document receive the new version of the document and are hence notified about the change.

There are two engineers (1,5 FTE) that co-ordinate the change control process, their tasks involve:

- Distribute the Change Notice and the related documents
- Keep track of the Change Notice's status
- Expedite disciplines for completion
- Liase with Project Controls for PCN and Internal/Contract Change order files and reports
- Expedite client for timely client decisions
- Advise disciplines about the rejection or approval of the change
- Maintain logs, and report status of PCN and Internal/Contract Change Orders.

Their main task is to monitor the Change Notices and make sure that progress is being made. When necessary they arrange meetings in which the Change Notices are discussed. Therefore, the Change Process Co-ordinators have an overview with respect to all the Change Notices that are being processed simultaneously.

6.3.6 Use of PDM at Chemical

In this section is discussed how ICT systems containing PDM functionality enabled starting on preliminary document versions.

6.3.6.1 History of PDM at Chemical

Mid 1994 Chemical started with a feasibility study on Document Management. By the end of 1994 it became clear that for the long term Chemical needed a Product Data Management solution. For the short term Chemical decided to start with Document Management. After evaluating several systems mid 1995, a commercially available PDM system was selected. The selected PDM system is customised to match the needs of Chemical. The first functions planned for implementation in the PDM system involved File Management and Revision Control. According to the PDM function model in section 2.2.2 it involved Document Vault Management and Document Version Control respectively.
As there was confusion about the exact procedure for version numbering and revision control it took longer than a few months as originally planned. By the end of January 1996 the customisation was completed and Chemical decided to conduct a pilot with the PDM system in one project. This required project specific adaptations of the PDM system. Firstly, all a complete list of project specific document types had to be created. This proved to be more difficult than expected, because some disciplines had difficulty specifying the document types although they have been using them for years. End 1996, two and a half years later, the PDM system was operational as a result of the first project.

Later, other projects followed and gradually the functionality was expanded and more users were trained on the use of the PDM system in their daily work. The right column of Table 10 shows a list of functions that was planned for the long term, this list was created in 1995. In the left column is indicated which PDM functionality is supported today in Chemical's PDM system.

Table 10 - PDM functionality at Chemical

Planned functionality long term (in 1995)	Functionality according PDM Function Model	Realised today (2000)
Document filing and archiving	Vault management	2
Document archiving and retention	Vault management	0
Document registration	Document identification and numbering	2
Document class management	Document structure modelling	2
Document relation management	Document structure modelling	1
Document authorisation	Access and authorisation	1
Document approval management	Document status and release control	1
Document status control	Document status and release control	2
Document revision control	Document version control	2
Document change impact notification	Document change impact analysis	0
Document change administration	Workflow management	0
Document change tracking	Workflow management	0
Document control reporting	Work tracking/reporting	1
Document distribution control	Workflow management	1

0 = not realised; 1 = partially realised; 2 = fully realised

The table shows that Chemical implemented the basic functionality of a PDM system, focussed on managing documents. Other (main) functions from the PDM Function model such as Product Management and Process/Project Management are hardly supported (yet).
The omission of Process/Project Management means that there is limited support for electronic workflows. This is an omission because it is expected that this function is necessary to successfully start on preliminary information. This does not mean that Chemical does not see the advantage of workflow functionality, on the contrary. However, it was not possible to implement this functionality earlier because the basic functionality was not yet in place.
At this moment they are looking at how they can implement workflow functionality. An earlier attempt was unsuccessful because it seemed to be difficult to define the processes correctly. It was for example difficult for the disciplines to explicitly define the distribution roles and schedules.

To summarise it took Chemical five years to customise their PDM system and use it for all their major projects. They experienced that a 'standard' PDM system needs customisation to deliver the desired functionality. Moreover, Chemical experienced that implementing a PDM system is difficult if the processes are undefined. There is consensus needed in regard to the processes before they can be implemented. Besides, processes can vary per project therefore project specific customisations are required. A possible explanation for the amount of customisation is that Chemical purchased their PDM system in 1995. At that time PDM systems were not as sophisticated as today's systems. Besides, according to the PDM consultants, the PDM system at Chemical is typically known for its long implementation period.

6.3.6.2 PDM and Document Release at Chemical

The document lifecycles at Chemical are not implemented as workflow, using workflow functionality, in the PDM system. At Chemical the PDM system is used to monitor the paper workflow by registering the status of each document version in the PDM system. Therefore if the status of a document version changes somebody has to manually update the status in the PDM system. An advantage of implementing the document lifecycle of documents, using workflow functionality, is that it takes care of routing and distributing the document versions.

At Chemical document versions are still routed manually. The following example illustrates that the current document release process is labour intensive.

Example of release process at Chemical

If a discipline wants to release a version of a set of documents for AFD they need to be approved first. In this case the discipline has to prepare a transmittal and changes the status of the version of these documents in the PDM system. The documents then appear in the worklist of the Project Document Control department, they print the document versions and the transmittal and distribute it to the reviewers.
The review process is often done sequentially which means that the Project Document Control distributes it to the first person. If this person finishes the review he changes the status of the document version in the PDM system. The Project Document Control then picks up the marked-up document version(s) and brings it to the next reviewer. In other words the Project Document Control department is responsible for expediting the documents in the review process.
If the document versions have been reviewed they can be released for AFD. Once again the discipline changes the status of the document versions in the PDM system. The documents then appear in the worklist of the Project Document Control department, they print the document versions and the transmittal and distribute both according to the distribution list.

In section 4.4 it was mentioned that it is important to track progress of all documents when starting on preliminary information. The design control complexity increases because small sets of documents need to be managed. In section 5.1 it was mentioned that a PDM system helps to manage this complexity. If the PDM system contains status information of documents it can provide the required overview. At Chemical status information is managed manually in the PDM system (see the example in the previous section). However, the PDM system only monitors the status of a single workflow, they use several other systems to monitor the status at work package or project level. One system that is used by all disciplines is called MPACS. The documents in the PDM system are linked to work packages in the MPACS system. This makes it possible to automatically transfer document status information from the PDM system to the MPACS system.
Because of this link they are able to generate reports containing the status of for instance all documents in a specific work package or work area. This information is important to manage the engineering process. If the status report shows that the documents are behind schedule the area or project manager can take appropriate action.

MPACS is used for project planning purposes. At the start of a project the first thing that needs to be done is to draw up a project schedule. Initially they produce a Work Breakdown Structure (WBS) and for each activity in the WBS the start and end date is determined resulting in a level 1 schedule. This schedule is detailed in several rounds resulting in a level 2 schedule and a level 3 schedule. All the activities including their start and end dates are entered into the MPACS system. Every discipline has to update the progress of its activities in MPACS so that the project planning and control department can generate overviews of the total project progress.
A lot of activities, but not all, involve the creation of documents. Hence the status of documents tells you something about the status of the corresponding activity. Therefore, the documents in the PDM system are coupled with the corresponding activities in the MPACS systems. In other words, if the status of a document in the PDM system changes it results in a change of progress of the corresponding activity in MPCAS. The progress of activities in MPACS is not updated immediately, however, every night the PDM system and MPACS are synchronised.

PIA is a system that is used by the Piping discipline to manage the lifecycle of the 7,000 Isometrics that need to be created. The status of an Isometric is used as an indicator for its progress. Therefore PIAS contains functionality to generate progress reports based on the status of Isometrics. Prior to PIAS the Piping discipline had to collect and process the progress data manually. The PIAS system drastically reduced the effort of creating status reports regarding the progress of Isometrics. Moreover, with PIAS it is possible to create an up-to-date status report on demand, which was not possible without PIAS. The PIAS system is synchronised with the PDM system. The status of Isometrics in the PDM system is manually synchronised with the status in PIAS. Chemical already recognised that the functionality of the PIAS system can be replaced by the PDM system, but until today they did not replace PIAS yet.

To summarise at Chemical it is important to track progress, *i.e.* status, of all activities and documents. The main source for this information is the PDM system. However, there are also other systems that contain status information that is not found in the PDM system. These systems have been built in the time that Chemical was not using a PDM system and they had to look for other solutions to manage their processes.

6.3.6.3 PDM and Change Control at Chemical

Monitoring the change control process

The change control process as described in section 6.3.5 is not supported by the PDM system yet. Currently the Change Control Co-ordinator manages the change control process using an Access database. It contains meta-data about the Change Notices including the status of the Change Notices. However, the complete file of a Change Notice, *i.e.* the Change Notice form and the affected documents, is managed on paper. Distribution of the file is done on paper instead of using electronic routing of the Change Notice by means of the workflow functionality of a PDM system.
The Change Control Co-ordinator was asked if he would expect any benefits if the PDM system was used to support the change control process. The major benefits that he expects involve a reduction in the administrative part of his job, *i.e.* distributing and archiving the Change Notice files. He does not expect that a Change Notice will be processed more quickly. The duration of processing a Change Notice is determined by finding a solution and updating the documents and is not determined by distributing the Change Notice. While only the latter

is supported by PDM functionality the application of PDM will not result in a substantially shorter processing time of Change Notices.

Change impact analysis

At Chemical the PDM system, nor any other system, is being used for conducting a change impact analysis. An experienced engineer determines the impact of a change, if the change affects several disciplines the change impact is discussed by a group of engineers in one or more meetings. Determining the impact of a change depends on the knowledge of the engineers that are involved in this process.

A PDM system can determine the change impact by finding the units or pieces of equipment that are affected. But a change is more than determining the change impact. One engineer mentioned that a change is a subject that needs to be discussed. An example is replacing the material of a particular piece of equipment. In such a case it is important to know whether or not it is possible to change the material, what are the cost of the material, and what is the availability of the material. To answer these questions the change impact analysis function of a PDM system cannot be used.

The change impact analysis function is only a small part of the total process of determining the impact of a change. The question is then, whether the change impact function of a PDM system substantially improves the change control process. According to some engineers a PDM system would help in being more accurate. In other words they would not overlook documents that needed to be updated. On the other hand this is not a major problem in the current situation.

Faster communication of changes

According to engineers at Chemical PDM functionality will not result in faster communication of changes. It is assumed that fast communication is important if activities start on preliminary information. After all if a change is communicated slowly it results in a lot of re-work because the organisation is informed too late.

When a change is initiated the impact of a change is determined. If the change has a major impact on other activities the employees executing these activities are notified immediately by giving them a phone call or an e-mail. Often it is not known which specific activity is affected by the change but it is known which discipline is affected. Therefore, the lead engineer of the affected discipline is contacted and he is able to tell which activity is affected. He will warn the engineer that is affected by the change and puts that activity on hold.

At a later stage the discipline will be officially notified about the change because they receive the Change Notice form and they are invited to join a meeting in order to discuss the change. A PDM system supports the formal change notification process, which is always slower than the informal phone call. Therefore, it is not expected that a PDM system will improve the communication speed.

6.3.7 Benefits of PDM at Chemical

In this section the benefits of PDM functionality at Chemical are discussed. At Chemical they use a commercial PDM system as well as in-house developed systems that contain PDM functionality.

6.3.7.1 Expected benefits of PDM for release

At Chemical they start on preliminary information, but they do not use PDM functionality to its full potential to support the preliminary release process. A hybrid environment is used that has been described in section 6.3.4. Document statuses are maintained in the PDM system but the versions of the documents are distributed on paper and expedited by the Project Document Control group. Despite the fact that this process is still labour intensive they already see some benefits. Due to the registering of the document status in the PDM system it is easier to track progress on individual documents. Moreover, the status information from the PDM system is transferred to a project control tool (MPACS) that generates status reports. Hence the whole process of collecting and processing document status data became less labour intensive. Moreover, the status reports are generated more often so that everybody has more up-to-date information.

But not all documents are managed in the PDM system, for instance the Isometrics. To manage the progress of these documents they have developed an in-house system that has the functionality that is described above. The use of this system results in similar benefits.

The lack of PDM functionality to support document routing in the release process represents a huge potential saving. In a large project, like the one that is studied, the Project Document Control group consists of 10-12 FTE. Using PDM functionality for routing documents the number of headcount can be reduced with 40-60%.

But the most important finding is that the interviewees indicate that the PDM functionality makes it possible to track individual documents or small sets of documents. Without this functionality it is hardly feasible or very labour intensive to start on preliminary information. The functionality is essential in order to make sure that the right document versions are available for Construction at the right moment.

A PDM system can be used to track who is working on preliminary document versions. At Chemical the PDM system is not used for this purpose. They still use hard copy distribution lists to inform people about a new version of a document. The findings in the case study showed that a PDM system is not required to communicate changes faster. Informal communication structures already guarantee that changes are communicated quickly, for instance by phone, and that activities are put on hold that started on preliminary information. However, there are also benefits because a PDM system will make it easier to distribute document versions. With a PDM system it can be done electronically instead of manually, resulting in a more efficient notification process.

Moreover, it is also possible to introduce new features such as subscribing to distribution list and a more precise distribution schedule. Today employees receive every instance of a document type instead of only the documents that they need. Today this is not possible because of the extra administration and co-ordination effort that would be required.

Currently they do not store a distribution list if a preliminary version of a document is released with status For Information or Preliminary (preliminary document release of category 3). With a PDM system it would be easier to log who has received a preliminary version of such a document. However, at Chemical they have decided that it is not necessary to keep those persons updated that received a preliminary document version that belongs to category 3 (unplanned, unchecked, and unregistered).

6.3.7.2 Expected benefits of PDM for change control

The PDM system at Chemical is not used for doing a change impact analysis. They plan to implement this function in the future. Moreover, at this moment there is no system that has similar functionality. This can mean that they do not need a system to support this function. Some material from the case study supports this conclusion. It was found that determining a change impact includes more than determining which pieces of equipment (*i.e.* plant items) and documents are affected by the change. Nevertheless some engineers think that PDM functionality might be useful for determining the change impact. It prevents overlooking documents that need to be updated.

However, there is no reason to assume that PDM functionality that supports the change impact analysis is important for starting on preliminary information. This does not mean that PDM cannot improve the change control process. On the contrary, but it will mainly result in efficiency benefits. A PDM system could for instance replace the current Access database system. Managing and routing the Change Notices electronically reduces the administrative work of the Change Control Co-ordinator. Moreover, it will also increase the transparency of the change process because everybody can for example look up the number of Change Notices and the status of these Change Notices. Finally, the Change Control Co-ordinator only manages the Change Notices until they are approved. After that it is the responsibility of the engineers to update the affected documents and to notify everybody about the change. With a PDM system one can easily track the complete lifecycle of a change.

6.3.8 Conditions for success

An important prerequisite with respect to realising performance improvements using PDM functionality is having all documents available in digital form. Otherwise it is not possible to manage and monitor the workflow electronically. In the case study it was found that there are some hurdles to take before all documents are available in electronic form. The following illustrates the problem of getting all documents available in electronic format.

Electronic availability of documents

The PDM system at Chemical meets the first requirement only to a certain extent because not all documents are electronically registered and available in the PDM system. The Piping Studies and the PEFS are two examples here. The Piping Studies are performed manually on the drawing board. These drawings are registered in the PDM system, but the drawing is not available electronically. However, only a relatively small group uses Piping Studies. Moreover, this group has the paper master copies at their disposal in a filing cabinet near their workplaces. Therefore, it is not essential to have them electronically available

The PEFS on the other hand are the basis for further engineering activities and are therefore used by a large group. Although the PEFS are registered and electronically available in the PDM system, the electronic file in the PDM system is not the master. The process engineers mark-up the changes on the hard copies of the PEFS. Hence the latest version of the PEFS is found in a filing cabinet near the process engineers instead of in the PDM system. To keep everybody updated, the Process discipline regularly sends copies of the paper masters to other disciplines. Later the drafting department processes the marked-up paper masters resulting in electronic copies of the PEFS that are checked-in to the PDM system. These PEFS are already outdated because the process engineers continue to mark-up the paper masters. Conclusion, the PDM system is used for registration afterwards.

Updating the electronic PEFS is not done very frequently because of the high costs that are involved. There are approximately 500 PEFS, updating the PEFS costs more than one thousand man-hours each time because draftsmen have to make the changes and the process engineers have to check the updated PEFS.

Electronic review and sign-off
There are two conditions that have to be met to make electronic workflows a success at Chemical:
- Electronic review
- Electronic sign-off

The first condition requires that there is an electronic review process in place. In a review it is quite common that the reviewer marks-up a paper copy of a document version, by indicating his changes on a paper copy of the document version. If mark-up functionality is not available engineers will print the document version and indicate their changes on this copy. Because the changes are made on a hardcopy of the document version it is not possible anymore to use the PDM system for the distribution of the marked-up document version. Once again distribution of the marked-up document version is done manually. Therefore the advantage of using PDM functionality vanishes.
At Chemical the document lifecycle is not supported by workflow functionality and mark-up and redlining functionality is not used either. However, some engineers were asked what they think of mark-up and redlining functionality. Their comments were that it is not workable to mark-up a A1 size document on a 15” screen. They have to zoom-in and zoom-out all the time. Zoom-in to see the details and make some changes and zoom-out to get an overview. It is expected that electronic review of documents is more time consuming than reviewing a hardcopy of a document.
The second condition involves electronic sign-off. Documents that are released need a signature that proves that the document is authorised. As long as an electronic signatures are not accepted (by law and/or customers) documents are printed, signed, and send to the supplier by surface mail instead of electronic mail.

6.3.9 Conclusions

The previous section presented empirical data to support/reject the hypotheses. In this section the observations are summarised resulting in the conclusions of this case study. The conclusions are structured according to the four hypotheses mentioned in section 5.4.

Hypothesis 1
Chemical formally starts on preliminary document versions as defined in section 5.4:
- releasing smaller sets of documents that have to be monitored and tracked,
- these smaller sets of documents are released more often as preliminary document versions, and
- people start on the preliminary document versions which increases the risk of re-work and delays

They use it to overlap different phases in the EPC process, for instance the overlapping of Engineering and Construction. This type of starting on preliminary information, *i.e.* interdomain start-finish relationships, was classified as category 1 in this chapter.

Within a phase, the Basic Engineering phase for instance, activities are not formally started on preliminary information. Early in the project engineers are figuring out what the requirements are and try to formulate the requirements for their part of the project. Moreover, they are trying to define some high level solution to fulfil these requirements. As one engineer stated it: "*Early in the project your are happy with every piece of information you can get*". In other words although it not possible to define the release of preliminary information there is the need to share information in a rather informal way instead of a formal way supported by document statuses. When engineers start on preliminary in this phase, *i.e.* intradomain start-finish relationship, it is often a preliminary release that was classified as category 2 or 3 in this chapter. Conclusion, Chemical starts on preliminary information in different ways in different situations.

The case study at Chemical also showed that starting on preliminary information can not be applied to all stages in the product development process. They do not start on preliminary information before the plant architecture is finalised. Before that time the uncertainty and the impact of changes on downstream processes is too high. This is in line with what was mentioned by Hanssen (2000), one should not start on preliminary when the uncertainty is high.

Hypothesis 2
In this thesis it is assumed that starting on preliminary information increases the control complexity of the product development process. At Chemical they indeed track the progress of small sets of document or even individual documents. Because of the volume of documents, for instance 7,000 Isometrics, the complexity of tracking the progress of documents increases. This is supported by the fact that they indicate that monitoring the progress of Isometrics (when starting on preliminary information) was not possible without the in-house developed system, PIAS.

Although the functionality is available in the PDM system they do not explicitly register the relationships between plant-items and between plant-items and documents at Chemical[20]. It was expected that this was important in order to define the impact of a change as quickly as possible. However, in the case study it was found that determining the change impact is considered to be a small activity in the whole change process. Moreover, they do not need to register the relationships explicitly because they can just tell which documents are affected. Otherwise this information is implicitly stored in documents because in the documents the plant-items that are connected/related can be found. Moreover, documents contain references to other documents.

When starting on preliminary document versions it is important to notify everybody as quickly as possible about a change. In order to be able to notify everybody as fast as possible it is assumed that there should be a register of who is working on a preliminary version of a document. At Chemical it is known who is working on a preliminary document version because when a preliminary version of a document is released they create a distribution list. The distribution list is filed together with the preliminary document version. In the case of a change they look up the distribution list and notify everybody on the distribution list.
In section 5.1 it was suggested that the output/input relations between documents should be modelled in the PDM system. In case a released version of a document changes all the activities, which use this released document version as input should be notified. At Chemical these relationships between documents are modelled implicitly. They use distribution lists that

[20] One exception was found: the relation between equipment items and equipment requisitions.

tell you who (a function/person) receive versions of documents of a specific type. The conclusion is that they do register input/output relationships between documents by means of distrubution lists to register these relationships, but that they do not use the PDM system for it yet.

Hypothesis 3
The following provides an overview of the PDM functionality that enables starting on preliminary document versions at Chemical. This is done on basis of the requirements that were formulated in section 5.2.

The following table provides an overview of the required PDM functions that are used at Chemical to support requirement 1.

Table 11 - Required PDM functions for requirement 1 at Chemical

Required PDM functions	Chemical
vault management	+
document identification and numbering	+/-
document viewing and mark-up	-
scanning and imaging	-
document structure modelling	-
document version control	+
document status & release control	+
document baselines	-
process identification	+
work breakdown structure modelling	+
work planning	+
work tracking	+
workflow management	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

The following table provides an overview of the required PDM functions that are used at Chemical to support requirement 2.

Table 12 - Required PDM functions for requirement 2 at Chemical

Required PDM functions	Chemical
vault management	+
document identification and numbering	+/-
document structure modelling	-
document structure viewing	-
document change impact analysis	-
document version control	+
product identification	+/-
product structure modelling	+/-
product structure viewing	+/-
product change impact analysis	-
product version control	+/-
workflow management	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

The following table provides an overview of the required PDM functions that are used at Chemical to support requirement 3.

Table 13 - Required PDM functions for requirement 3 at Chemical

Required PDM functions	Chemical
vault management	+
document identification and numbering	+
document version control	+
document status & release control	+
(workflow management)	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

From these tables can be concluded that the basic PDM functionalities are in place and the major omission is the support workflow management, which is a requirement to enable starting on preliminary information. On the other hand they have good PDM functionality in place for monitoring progress on document level as well as activity and project level.

Hypothesis 4

At Chemical they use hybrid environment, document statuses are maintained in the PDM system but the versions of the documents are distributed on paper and expedited by the Project Document Control group. Due to the registering of the document status in the PDM system it is easier to track progress on individual documents. Moreover, the status information from the PDM system is transferred to a project control tool (MPACS) that generates status reports. Hence the whole process of collecting and processing document status data became less labour intensive. As a result it is also easier to generate status report. But the most important finding is that the interviewees indicate that the PDM functionality makes it possible to track individual documents or small sets of documents. Without this functionality it is hardly feasible or very labour intensive to start on preliminary information.

The lack of PDM functionality to support document routing in the release process represents a huge potential saving. In a large project, like the one that is studied, the Project Document Control group consists of 10-12 FTE. Using PDM functionality for routing documents the number of headcount can be reduced with 40-60%. But before these potential savings can be realised there are also conditions that have to be met. Firstly, it is mentioned that all documents have to be available electronically. Secondly the review process requires mark-up and redlining functionality and in some cases larger monitors. If these conditions are not met, a company will fall back on a (partly) paper based release process like Chemical.

The PDM system at Chemical is not used for doing a change impact analysis. Nevertheless some engineers think that PDM functionality might be useful for determining the change impact. It prevents overlooking documents that need to be updated. Instead of supporting the change impact analysis they believe that it is better to use PDM functionality to improve the whole change control process A system with PDM functionality could for instance replace the current Access database system. Managing and routing the Change Notices electronically increases the efficiency. However, it is not believed that PDM functionality results in faster notification. With a PDM system it can be done electronically instead of manually, resulting in a more efficient notification process. The most important benefit is that it increases the transparency of the change control process resulting in better control of engineering changes over their complete lifecycle.

Other findings

This section describes several other findings that are relevant for this research. The first finding is that some employees at Chemical object to releasing preliminary document versions because it means that several preliminary document versions have to be distributed instead of only the final version. Distributing several preliminary document versions of a document is time consuming. Therefore, they postpone the release of preliminary document versions if that is possible. It was assumed that a PDM system could remove this obstacle because with a PDM system document versions can be distributed electronically what makes the release of preliminary document versions less time consuming. But the manual distribution of preliminary document versions is not the only bottleneck. Before preliminary document versions are released they have to be checked. This is extra work that cannot be eliminated by a PDM system.

However, it was assumed that the release of a preliminary document version would require less checking because the document version is released for a specific purpose. Therefore, one would expect that the preliminary document version is only checked on this aspect. However, at Chemical a preliminary document version undergoes the same checks as the final version of a document. Once a document version is preliminarily released engineers have to keep a log of all the changes. When the next version is preliminarily released the log of all the changes is attached to the new version. This log tells the receiver of the new version what the differences are with respect to the previous version of the document.

A PDM system does not make it necessarily easier to release preliminary document versions. It simplifies the distribution process but it does not eliminate other tasks such as checking and recording changes. Therefore one should be selective with respect to the release of preliminary documents versions. A good criterion is that only versions of documents on the critical path should be released preliminary, because only it will result in a reduction of the lead-time. However, this means that documents of the same document type can have a different release process. It depends on whether or not they are on the critical path.

The second finding concerns how a preliminary document version is released. In my theory it is defined that a preliminary document version can be released for a specific purpose. This means that a specific aspect of the preliminary released document version is frozen. The case study showed that it is also possible to release a selection of objects in a document. The other objects in the document are not released and put ‘on hold’.

If a document version is frozen it normally means that it is not allowed anymore to make changes to the document version. However, if a preliminary document version is released only the aspects or objects that are unfrozen can be changed. With the current IT systems it is possible to freeze a complete document version but it is very hard to freeze an aspect or a number of objects of a document version, but there are some exceptions.

7 Conclusions

In this chapter the conclusions are presented. There is a section for each hypothesis that is mentioned in section 5.4. This is followed by a discussion concerning the results of the research and a direction for future research.

7.1 Hypothesis 1: Design of the release process

The first hypothesis predicts how companies start on preliminary document versions. The case studies show that both companies do start on preliminary document versions as predicted. However, both companies did not formalise their procedures for starting on preliminary document versions in their quality systems yet. Over the years they gradually developed a working practice for starting on preliminary document versions. One (external) driver for developing these working practices is the pressure to reduce the cycle time of projects.

The case studies also led to further refinement of the theory on starting on preliminary document versions, it concerns:
1. the flexibility of the release process
2. the application of starting on preliminary document versions

Ad1) Flexibility of the release process
In section 4.3.4.2 a document lifecycle is presented that includes preliminary release steps. The presented document lifecycle assumes that all documents of a certain type will go through all the defined preliminary release steps (i.e. each release step is a new version of the document). The case studies revealed that a more flexible document lifecycle or release process is required.
The Chemical case study showed that they have a *planned* and an *unplanned* release process. A *planned release process* means that it is possible to define all the preliminary release steps of a document beforehand. This means that all instances of a certain document go through all the defined (preliminary) release steps. An *unplanned release process* means that not every document goes through all the pre-defined release steps. In other words two documents of the same type can have different release steps. This suggests that it should be possible to skip certain release steps when necessary. In the *unplanned release process* Chemical has the possibility to define new (ad-hoc) release steps. For example, during a project it might be necessary to release a version of a document for a specific purpose that is not foreseen. In that situation a new (preliminary) release step is added to the document lifecycle of that particular document version.

Ad 2) Application of starting on preliminary document versions
My theory on starting on preliminary document versions as presented in this thesis is not very specific about the situation in which the technique of starting on preliminary document versions should be applied. But the case studies provided clear indications concerning its application in certain situations. It should be applied to:
– activities that are on the interface of two domains, and
– on the critical path of the project.

My theory assumes that starting on preliminary document versions can be applied at several levels, within domains (intra domain relationships) and between domains (inter domain relationships). However, in both companies starting on preliminary document versions is only used on the interface between engineering and production, *i.e.* on the interface of two domains. Starting on preliminary document versions is less common if it involves activities within the same domain. One of the reasons is that it concerns *creative iterations* within a domain and between domains it concerns mainly *dysfunctional iterations*.

Creative iterations are often very short loops, hence a formal release process would slow down the process too much because of the extra administrative effort. Moreover, within a domain there is often a team that creates the product model for that domain. Such a team facilitates informal communication, which makes a formal release process superfluous. Between domains, on the other hand, it involves dysfunctional iterations that have a negative effect on cycle time and cost. A more formal procedure is required to control these negative effects. An example of an interface between two domains is the interface between engineering and production. The start of production means that the expenditures will rise significantly. Changes to preliminary released information will then result in a high cost. Therefore both companies find it important to define a formal process that describes when and which documents can be exchanged preliminarily.

There was also one exception for using a formal procedure for starting on preliminary document versions within a domain. This is the case when part of the work is outsourced. When work is outsourced it is often done at an hourly basis which means that one has to pay for each hour. In such a situation iterations result in extra cost because the iterations will result in extra hours of work. Therefore, it is important to define when which preliminary document versions are exchanged.

The case studies revealed that there is resistance against starting on preliminary document versions, which was not foreseen in my theory on starting on preliminary information. Each time a preliminary document version is released it needs to be checked, copied and distributed. Although PDM functionality can eliminate the copy and distribute activities it cannot eliminate the check activity. Moreover, all changes with respect to the (preliminary) released document versions have to be recorded. This is also a task that can not be eliminated by PDM functionality. In other words starting on preliminary information introduces extra administrative work for engineers.

This observation led to the conclusion that not all activities in a project should start on preliminary document versions. Only the activities on the critical path should be overlapped by starting on preliminary information because only that suffices to have the desirable result, *i.e.* a decrease in project cycle time. Overlapping activities that are not on the critical path will only have negative effects. It increases the administrative effort and it does not decrease the project cycle time.

7.2 Hypothesis 2: Manage control complexity

In the second hypothesis I made the assumption that starting on preliminary document versions, increases the control complexity. To manage the control complexity a company should:

1. track progress, *i.e.* status, of small sets of documents or individual documents,
2. register relationships between components, documents as well as components and documents, and
3. register who is working on basis of a preliminary version of a document so that they can be notified in case of a change.

According my theory each company that starts on preliminary document versions has to conduct these activities.

Track progress
Both companies track the progress of individual documents or small sets of documents. At Aero this is done manually in an Excel spreadsheet. At Chemical they use different IT systems that contain PDM functionality. Both companies indicate that tracking individual documents or smaller sets of documents has increased the control complexity of the document release process.

Register relationships
The case studies did not provide any proof that companies starting on preliminary information register relationships between parts, documents and parts and documents to determine the change impact. Engineers are able to do a change impact analysis on the basis of their knowledge, *i.e.* they construct a virtual model of the relationships in their head. Even if a product consists of thousands of parts they can still rely on their knowledge. Hence there is no urgent need to capture these relationships in a spreadsheet or a database.
However, engineers at both companies also mention that they do not capture the relationships because it takes a lot of effort to capture and maintain these relationships. But they indicate that if they would have the proper tools they might consider registering the relationships because they see the added value of it.
Both companies indicated that the change impact analysis is only a small part of the complete change control process. It is more important to monitor progress of changes and to make sure that the changes are implemented in time. Both companies use Excel spreadsheets or databases to track the progress of changes.

Notification
Only one of the two companies registers who is working on a preliminary document version by maintaining a distribution list. The other company only maintains a distribution list for the release of the final version of a document. The main reason is that the distribution of preliminary document versions is straightforward. There is one production specialists who receives ***all*** preliminary document versions and he takes care of further distribution within his department. At the other company this is not possible because of the size of the project and the number of documents involved.
Both companies use a two step distribution schedule. A document version is distributed to lead engineers or discipline leaders that take care of further distribution of the document version within their group. They agree that this is a comprehensive way of working but argue that maintaining distribution lists on document or document version level is not feasible because it would increase the required administrative effort.

To conclude this section the complexity of both case study companies is presented. In section 5.5 is mentioned that PDM does not act as an enabler for starting on preliminary document version if the complexity of the product development process is low. Four factors have been identified which determine the complexity of the product development process: product complexity, product modularity, number of changes, and geographical spread.

Although it is perhaps possible to determine an absolute value for these complexity factors it is hard to determine at what values PDM is required as an enabler. Therefore a relative score is used to position both companies against each other. Table 14 shows the outcome of comparing the complexity factors at both companies.

Table 14 - Comparison of complexity factors at Aero and Chemical

Company	Complexity of product	Modularity of product	No. of Changes	Geographical Spread
Chemical	high	low	high	low
Aero	low	low	low	low

Chemical and Aero differ on the aspect of product complexity and number of changes. The conclusion is that the complexity at Chemical is higher then at Aero because the product complexity and the number of changes are higher. This might explain why Aero is using less advanced systems for tracking progress than Chemical and also does not register the distribution lists of preliminary released document versions.

7.3 Hypothesis 3: Enabling capability of PDM

In this section the applied PDM functionality of both companies is reviewed. The case study results can be summarised in Table 15, which is a combination of the tables presented in section 6.2.6 and 6.3.6. The left column of the table is the PDM functionality that can act as an enabler for starting on preliminary document versions according my theory. The two columns on the right side indicate the PDM functionality that is used by Aero and Chemical. As mentioned before this does not necessarily mean that the functionality is implemented in a PDM system, it also includes PDM functionality that is implemented in in-house developed systems.

Table 15 - Comparison of PDM functionality applied at Aero and Chemical

PDM functions	Aero	Chemical
vault management	+	+
document identification and numbering	+/-	+/-
document viewing and mark-up	-	-
scanning and imaging	-	-
document structure modelling	-	-
document structure viewing	-	-
document change impact analysis	-	-
document version control	+	+
document status & release control	+	+
document baselines	-	-
product identification	+/-	+/-
product structure modelling	+	+/-
product structure viewing	+	+/-
product change impact analysis	-	-
product version control	+	+/-
process identification	+	+
work breakdown structure modelling	+	+
work planning	+	+
work tracking	-	+
workflow management	-	-

Legend: + = enabled by PDM functionality; - = not enabled by PDM functionality

The table only shows whether the companies use a specific PDM functionality and is not an indication of the quality of the PDM functionality. Although it is fair to mention that the PDM functionality at Chemical is on average more advanced than the PDM functionality at Aero.
From the table it can be concluded that both companies have implemented similar functionality. For this functionality it seems justified to conclude that it is required to support starting on preliminary information. But there is also functionality that is not implemented by the case study companies. For example the functionality that is required for doing a change impact analysis such as document structure modelling, document structure viewing, document change impact analysis, product structure modelling, product structure viewing, and product change impact analysis. The question is whether my predictions were wrong or that there is another explanation. On basis of the case study results it is possible to conclude that the latter is the case. The companies do not register relationships between parts, documents and parts and documents to determine the change impact. Engineers are able to do a change impact analysis on the basis of their knowledge. However, engineers at both companies indicate that if they would have the proper tools they might consider registering the relationships because they see the added value of it.
Another example of PDM functionality that is not used by the case study companies is document viewing and mark-up and scanning and imaging. The reason for this is that they do not manage all their documents in the PDM system yet and still rely on paper-based processes for reviewing and distributing documents. As long as they do not manage all their documents in the PDM system and adapt their process for reviewing documents it is likely that they will not use document viewing and mark-up and scanning and imaging functionality. Moreover, as long as they have not implement this functionality they will not be able to take full advantage of workflow functionality.

A plausible explanation for the observation that both companies did not implement all required PDM functionality is that they are in the middle of a change process. It involves the change from a sequential development process to a concurrent development process. Such a change is not realised in a few weeks; it is more likely that it will take months or even years. In this change process it is possible to identify several intermediate states. This is also the idea behind so-called maturity models such as the Capability Maturity Model for Software Development, Capability Maturity Model for Integrated Product Development (CMM-IPD) and the Readiness Assessment for Concurrent Engineering (RACE).
The RACE model that has been further developed by De Graaf (1996) into RACEII can be used to determine the maturity level of an organisation (as-is) on two aspects: process and technology (ICT). Each aspects consists of several dimensions that are scored on a scale from 0 - 5. Based on the as-is maturity level the RACE method helps to define the to-be maturity level of an organisation. If a company wants to implement ICT functionality, the maturity of this functionality should match the maturity of the business processes. For example, it does not make sense to implement sophisticated planning functionality if a company does not have a proper planning process.
This knowledge can also be applied to PDM functionality and starting on preliminary information. If the maturity of the organisation with respect to starting on preliminary information is low then also the maturity of the applied PDM functionality is also low. This aspect was not taken into account during the selection of the case studies. However, the opinion of two experts has been used to select two companies in the Benelux. Therefore there is no reason to belief that one can find another company with a higher maturity with respect to starting on preliminary information and corresponding higher maturity of PDM functionality.

7.4 Hypothesis 4: Performance improvements

In this section the performance improvements are discussed that follow from the use of PDM functionality to enable starting on preliminary information. The performance improvements are grouped according the requirements that have been formulated in section 5.2.

Requirement 1
PDM functionality that supports requirement 1 should result in the following performance improvements:
1. Less administrative effort to track the lifecycle of documents
2. Less administrative effort to release (*i.e.* copy and distribute) a version of a document
3. Instant overview of the actual status of documents, activities, and project

Ad 1) Both companies invest time in tracking the lifecycle of individual documents, *i.e.* track progress. For example, Chemical has to track thousands of documents during a single project. Therefore they developed systems with PDM functionality to track the lifecycle of these documents. They claim that these systems reduced the administrative effort to collect data when comparing it to the situation where this data was collected manually.

Ad 2) Both companies spend a lot of time on copying and distributing documents when (preliminary) releasing document versions. In the case of Chemical there is even a separate department that takes care of copying and distributing document versions. Both companies indicated that the suggested implementation of PDM functions, especially workflow management functionality, will contribute to a reduction in the effort that is required for copying and distributing document versions. In the case of Chemical it means a saving of several FTE on one single project.
However, the release of preliminary document versions also raises the cost, because it requires extra checking and registering changes once a version is preliminarily released. This work can not be eliminated by a PDM system. At Chemical there was an extra hurdle because process engineers do not make drawings themselves (see section 6.3.6.2). Each time the PEFS are released they send an update of the paper master copy to the draftsmen who update the electronic copy. The process engineers then check the work of the draftsmen, in the studied project it led to approximately 1,000 hours of extra check work each time the PEFS are released.

Ad 3) Both companies have an overview of the actual status of documents in their lifecycle because they systematically collect and register this data. They both indicate that the suggested implementation of PDM functionality, especially workflow management functionality, will result in less effort to collect the required data because it can be collected automatically instead of manually. But the main benefit is that it increases the transparency of the process and that everybody has instant access to the status of the project. At this moment information is collected with certain intervals (every 2 or 4 weeks) and then published on the Intranet or distributed via e-mail.

Requirement 2
PDM functionality that supports requirement 2 should result in the following performance improvements:
1. Faster impact analysis
2. Better quality of impact analysis, not overlooking things
3. Less administrative effort to route/distribute and monitor the change process
4. Instant overview of all pending changes and their status

Ad 1 + 2) Both companies do not have a change impact analysis process in place as was expected therefore they also do not use PDM functionality to support this process. In both cases the change impact is determined by experienced engineers during a meeting in which they analyse the change. There were no examples that this method of determining the change impact leads to mistakes or errors therefore it is assumed that a PDM system has no added value here. Moreover, it is indicated that they do not expect that PDM functionality will speed up the process of change impact analysis.

Ad 3) Both case study companies have one or more co-ordinators that take care of routing and distributing the change requests and monitoring the change process. Both case study companies claim that a PDM system will reduce the effort that is needed to manage the change process. But on the other hand they do not expect that it will speed-up the complete change process because a large part of the cycle time consists of queuing time (*i.e.* the document version is waiting in somebody's inbox).

Ad 4) At Chemical only the change co-ordinator has access to data regarding the progress of pending changes. At Aero they publish a spreadsheet on the network with certain intervals so that everybody has access to the latest status of the pending changes. A PDM system will eliminate the effort involved in updating the database/spreadsheet manually. Moreover, it will eliminate the effort involved in publishing the information because every project member has direct access to this information. Another benefit is that progress information is no longer published at certain intervals. Instead project members will always have access to the current status of pending changes.

Requirement 3
PDM functionality that supports requirement 3 should result in the following performance improvements:
1. Less administrative effort for maintaining distribution lists
2. Faster notification and hence less re-work
3. Possibility to have a customised distribution list for every individual document or document version

Ad 1) At Aero they do not maintain a distribution list of the preliminarily released document versions only of the final release of a document version. The group of people that receives a preliminary version is so small, *i.e.* the complexity is that low, that it is known who received a preliminary version of the document. Therefore, they hardly put any effort into maintaining a distribution list. At Chemical the complexity is higher and therefore they maintain a distribution list for preliminary released documents. Although they did not implemented the predicted PDM functionality they agree that it will enable them to automatically register all releases of preliminary information resulting in a reduction of administrative work.

Ad 2) The application of PDM functionality will not lead to faster notification of changes. Currently project members are informed because they (or their supervisors) participate in meetings where the change is discussed. A PDM system will never be faster than this informal oral communication. Hence applying PDM functionality for notification purposes will not considerably reduce the amount of re-work.

Ad 3) At this moment both companies use ‘coarse’ distribution lists on document type level. They are not able to maintain distribution lists on document or document version level because it requires too much administration. The implementation of the suggested PDM functions will enable them to maintain distribution lists on document or document version level without having the administrative burden of archiving the distribution lists. Moreover, a PDM system offers employees the possibility to subscribe/unsubscribe themselves to a distribution list on a document level. Engineers from the case study companies agree that this would enable them to maintain distribution lists on document or document version level.

Summarising

Both companies indicate that the application of PDM functionality will not lead to a faster change impact analysis, a better quality of the change impact analysis process or a faster notification of changes. Therefore it is not justified to claim these benefits as a result of applying PDM functionality as enabler for starting on preliminary document versions.

The other claims concerning the application of PDM functionality benefits are all supported by the findings in the case studies. Amongst others it results in efficiency benefits, such as the elimination of copy, distribution, register and archive activities in the release and change control process. But the main contribution is that it increases the transparency of the process. This benefit is driven by the application of workflow functionality in combination with document management functionality. The increased transparency makes the whole release and change process better manageable. However, both companies did not implement this functionality yet.

The support of PDM functionality is only required in complex projects. In projects of low complexity one person can keep the required overview and can co-ordinate all release and change control activities. But when it involves projects where dozens of people have to co-operate and it involves several hundreds of drawings or more PDM functionality enables companies to start on preliminary document versions.

7.5 Discussion

This section contains a discussion regarding the results of the research and the possibility to generalise these results. The main result is a theory on starting on preliminary information. It describes the processes that are required to start on preliminary information. This resulted in a guideline for the design of the document release process. To describe the document release process I have extended the available knowledge on document version, document status, document lifecycle and document workflow. This resulted in a method to describe/model the process of starting on preliminary information more precisely. I have also shown how this modelling method can be used to model the complete product development process using the notion of domains and product decomposition. In the RapidPDM project this model has been used to built a simulation model of product development processes. In this research the conceptual model helped to formulate the predictions (hypotheses) of the effect of PDM functionality with respect to starting on preliminary document versions. The predictions concern the activities that can be enabled using PDM functionality, what kind of PDM functionality is required, and the performance improvement that can be expected from

applying PDM functionality. Two case studies have been conducted to collect empirical material to validate the hypotheses. This validation is used to validate the whole theory on starting on preliminary information. The first four sections of this chapter have been used to discuss to what extent the case studies provide empirical data to support the hypotheses.
This led to a further refinement of the theory on starting on preliminary document versions, *i.e.* starting on preliminary document versions is applied in certain situations and requires a more flexible approach. The assumption that starting on preliminary information increases the control complexity is also justified because the case study companies conduct the predicted activities. To support these activities the companies use part of the PDM functionality that was predicted. The reason that they do not use all the predicted PDM functionality is attributed to the maturity level of both organisations. Finally the benefits of applying PDM functionality were predicted. The case studies show that not all the benefits are justified and the theory is changed with respect to this point. Summarising the case studies resulted in a validation and further refinement of the theory on starting on preliminary information and the enabling role of PDM.

The knowledge that has been developed in this research is not limited to the two case study companies that were involved in this research. As mentioned in section 1.4.2 the goal of the research is to achieve *analytical generalisation* by generalising the results of the case study into a theory that can be applied at other companies as well. In this research the theory has been built on basis of literature and is validated on four points using the two case study companies. The theory has a descriptive nature instead of a prescriptive nature meaning that it is possible to describe how other companies start on preliminary document versions and make predictions about the required PDM functionality and the performance improvements that can be achieved. An important factor that influences these predictions is the maturity level of an organisation. However, this was not taken into account in this research and therefore requires further research.
The domain to which my theory can be applied has already been defined in section 1.2, it involves product development processes with a low sensitivity (large changes in the exchanged preliminary information have a small effect on the downstream process) and a slow evolution (major changes happen until late into the upstream process). I only excluded the product development processes that were classified as breakthrough innovation processes. But even if breakthrough innovation processes are excluded it involves a very broad domain. To check if the theory is indeed applicable to a large range of companies a number of independent variables have been defined. It involves:

- the industry/market in which the company operates
- the product that is developed by the company
- the production method of the company, *e.g.* mass production or One-of-a-Kind

The two case studies that have been selected differ on all three independent variables. The case studies did not provide any indication that the results at both companies differed because of the different independent variables. The only differences that were found have been explained using the dependent variables:

- product complexity
- modularity
- number of changes
- geographical spread.

It resulted in the conclusion that Aero uses less advanced PDM functionality to control the complexity of the document release process than Chemical because the complexity of the product development process at Aero is lower. Therefore the conclusion is that this research resulted in a theory to describe the process of starting on preliminary document versions, make predictions about the required PDM functionality and make predictions about the performance improvements that can be achieved. Moreover, this theory is applicable to all companies in the specified research domain.
However, if a company in the research domain is looking for ways to reduce cycle time it should not only focus on the theory that is developed in this research. It should also take into account the other strategies that have been mentioned in section 1.1.

7.6 Future research

In this final section three directions for further research are discussed.

1. Further validation of case study results
The case study companies indicated that they are convinced of the use of PDM functionality as enabler for registering part/document relationships for change impact analysis and for maintaining distribution lists on document or document version level. However, at the moment they are not conducting these activities, not even manually, therefore it was not possible to test this conclusion. Further research should concern the development of PDM functionality to support change impact analysis and distribution lists on document or document version level. This functionality should be tested in a company in the specified research domain to see if the expectations with respect to this functionality are justified.
Another option is to apply the current research to other cases to achieve *literal replication*. The results of this research can be used to predict which other companies might also benefit from the results of this research. These case studies can be used for further validation of the research results. However, a more interesting question is to what extent the developed theory on starting on preliminary document versions can be used in breakthrough innovation processes.

2. CMM for starting on preliminary information
One of the conclusions is that the maturity of the organisation determines the maturity of the PDM functionality that can be applied and hence the performance improvements that can be achieved. Further research should develop a Capability Maturity Model for starting on preliminary document versions. Such a model should assist companies in increasing the overlap of their product development process in several (controlled) steps. While starting on preliminary document versions is a form of Concurrent Engineering the RACE model can be used as a starting point.
Another option is that the CMM model not only focuses on starting on preliminary information but that it also includes other techniques that can be used to reduce cycle time of product development processes. After all starting on preliminary information is not the only and also not always the best method for reducing cycle time of the product development process.

3. Implications of the shift from document to object management
Today the product development processes of a lot of companies are still document oriented. The Form, Fit and Function of the product is still captured in documents. However, this is changing because companies increasingly use 3D design tools in their development process. An example of such a tool is the PDMS tool at Chemical that is used for 3D-plant design.

These systems are not document oriented anymore but object oriented. The output of the design process is therefore an object model instead of a set of documents. This requires a different approach to starting on preliminary information, which is illustrated using an example.
After the release of the final version of a document the complete content is frozen, in other words the version of that document is put under change control. The PDM system freezes the document by locking the corresponding file. In that way the PDM system can prevent that the frozen document version is changed without the proper authorisation. In case of an objectmodel the PDM system should not freeze the complete object model but only certain objects that are released and therefore need to be frozen. This requires that the PDM system can manage the objects in a 3D model. Further research should investigate the organisational aspects as well as the technological aspects that are related to this shift from 2D design to 3D design with respect to starting on preliminary information enabled by PDM.

References

[Albano *et al.*, 1993] Albano L.D., Conor J.J., Suh N.P., 'A framework for performance-based design', Research in Engineering Design, vol. 5, 1993, p.105-119.

[Andreasen & Hein, 1987] Andreasen M.M., Hein L., ***'Integrated Product Development'***, Springer, Berlin, 1987.

[Blackburn, 1991] Blackburn J.D. (Ed.), ***'Time Based Competition: The Next Battleground in American Manufacturing'***, Business One Irwin, Homewood, 1991.

[Boothroyd et al., 1994] Boothrroyd G., Dewhurst P., Knight W., ***'Product design for manufacture and assembly'***, Dekker, New York, 1994.

[Bos *et al.*, 1987] Bos, Terleth, Hoek van de, 'Vrijgeven en archiveren', Publication of the PMS project by Fokker/Nevesbu/Rijkswaterstaat, 1987 (In Dutch).

[Breuls, 1997] Breuls P., personal communications on PDM function framework, 1997.

[Breuls, 1999] Breuls P. in Korbijn A., ***'Vernieuwing in productontwikkeling'***, CIP-data Koninklijke Bibliotheek, Den Haag, 1999 (in Dutch).

[Booz, Allen & Hamilton, 1982] Booz, Allen, Hamilton, 'New product management for the 1980s', New York, 1982.

[Carter & Baker, 1992] Carter D.E., Stilwel Baker B., 'Concurrent Engineering: the product development environment for the 1990s', Addison-Wesley, Reading, USA, 1992.

[Chen & Tsao, 1998] Chen Y-M. and Tsao T-H., '***A structured methodology for implementing engineering data management***', Robotics and Computer Integrated Manufacturing, No. 14, p. 275-296.

[CIMdata, 1995] CIMdata, Inc., ***'Product Data Management: the Definition'***, http://www.cimdata.com, 1995.

[CIMdata, 1997] CIMdata, Inc., ***'Product Data Management'***, Report for the UK Department of Trade and Industry, http://www.dti.gov.uk/ (search for CIMdata), 1997.

[CIMdata, 1999] CIMdata, Inc., ***'Product Data Management: Market Pricing Analysis Report'***, 1999.

[CIMdata, 2000] CIMdata, Inc., Conference Proceedings of PDM Conferences from 1997-2000.

[Clark & Fujimoto, 1991] Clark K.B. and Fujimoto T., ***'Product Development Performance: Strategy, Organization and Management in the World Auto Industry'***, Boston: Harvard Business School Press, 1991, p.78.

[Clausing, 1994] Clausing D., ***'Total Quality Development: A Step-By-Step Guide to World-Class Concurrent Engineering'***, ASME Press, New York, 1994.

[Coolen, 2000] Coolen, T., ***'Development of a Simulation Model for the RapidPDM project'***, graduation thesis Eindhoven University of Technology, 2000.

[Cooper, 1993] Cooper R.G., 'Winning at new products: Accelerating the process from idea to launch', Perseus Books, Cambridge, USA, 1993.

[Cross, 1994] Cross N., ***'Engineering Design Methods: Strategies for Product Design'***, John Wiley & Sons , 1994.

[Dataquest, 1990] Carey J. and Schiller Z., ***'A smarter way to manufacture'***, Business Week, April 30, 1990 adapted from research from Dataquest.

[Davenport & Short, 1990] Davenport T.H., Short J.E., '***The New Industrial Engineering: Information Technology and Business Process Redesign***', Sloan Management Review, Summer 1990.

[De Graaf, 1996] Graaf R. de***, 'Assessing Product Development: Visualizing Process and Technology Performance with RACE'***, dissertation Eindhoven University of Technology, 1996.

[De Groot, 1994] Groot A.D. de, ***'Methodologie: Grondslagen van het onderzoek en denken in de gedragswetenshappen'***, Koniklijke Bibliotheel, Den Haag, 1994 (In Dutch).

[De Kok *et al*., 1995] Kok A. de, Mandemaker T.H.. and Cornelissen M., ***' PDM Selection Guide -From needs to selection; a business solution'***, Dutch Society of Informatics, May 1995.

[De Leeuw, 1996] Leeuw A.C.J. de, 'Bedrijfskundige Methodologie: Management van onderzoek', Van Gorcum, Assen, 1996 (In Dutch).

[Den Hertog & Van Sluijs, 1995] Hertog F. den, Sluijs E. van, ***'Onderzoek in organisaties: een methodologische reisgids'***, Van Gorcum, Assen, 1995.

[Doblies, 1998] Doblies M., '***Globales Produktdatenmanagement zur Verbesserung der Produktentwickelung***', Disseration, Techn. Univ. Berlin, 1998.

[DoD, 1997] US Department of Defense, ***'Military Handbook: Configuration Management Guideline'***, MIL-HDBK-61, US Department of Defense, 1997.

[Dorst, 1997] Dorst K., "***Describing Design - A comparison of paradigms***", Vormgeving Rotterdam, 1997, ISBN 90-9010822-X.

[Eppinger *et al.*, 1997] Eppinger S.D., Krishnan V., Whitney D.E., "***A Model-Based Framework to Overlap Product Development Activities***", Management Science, vol. 43, no.4, 1997.

[Erens, 1996] Erens F.J., '***The Synthesis of Variety: Developing Product Families***', dissertation Eindhoven University of Technology, 1996.

[Finkelstein & Finkelstein, 1983] Finkelstein L., Finkelstein A..C.W., ***'Review of Design Methodology'***, IEE Proceedings, 1983.

[Floyd *et al*., 1993] Floyd T.D., Levy S., Wolfman A.B., 'Winning the product development battle', IEEE, New York, 1993.

[French, 1985] French M.J., ***'Conceptual Design for Engineers'***, Design Council, London, 1985.

[Gartner Group, 1997] Burdick D., ***'CAPEII: Extending Engineering Competitiveness Acrosse the Supply Chain'***, Strategic Analysis Report - CIM, R-CAPE-127, Gartner Group, 1997.

[Gartner Group, 1998] Burdick D., '***PDM 1997 Year in Review***', Research Note- CIM, M-03-5341, Gartner Group, 1998.

[Goossenaerts, Pels *et al*., 1998] Goossenaerts J., Pels H.J. et al., ***'Product related data and knowledge management in the intelligent enterprise'***, Proceedings of IMS-Europe, Lausanne, April 15-17, 1998.

[Ha & Porteus, 1995] Ha A.Y., Porteus E.L., '***Optimal Timing of Reviews in Concurrent Design for Manufacturability***', Management Science, vol. 41, no. 9, 1995.

[Hales, 1991] Hales C., ***'Analysis of the engineering design process in an industrial context'***, Eastleigh, UK, 1991.

[Hamer & Lepoeter, 1996] Hamer P. van den, Lepoeter K., '***Managing Design Data: The five Dimensions of CAD Frameworks, Configuration Management and Product Data Management***', Proceedings of IEEE, Vol. 84, no. 1, 1996.

[Hameri, 1998] Hameri A.P. and Nihitilä, "***Product Data Management - exploratory study on state-of-the-art in one-of-a-kind industry***", Computers in Industry, Vol. 35, No. 3, 1998, p. 195-206.

[Hammer, 1990] Hammer M., ***'Reengineering work: Don't automate, obliterate'***, Harvard Business Review, July-August 1990, page 105-112.

[Hanssen, 2000] Hanssen R., ***'Concurrent Engineering vanuit beheersingsperspectief'***, dissertation Eindhoven University of Technology, BETA Research Institute, 2000 (In Dutch).

['t Hart, 1997] Hart M.W. 't, ***'Designing IT-support for Professionals'***, dissertation Eindhoven University of Technology, 1997

[Harris *et al.*, 1997] Harris S.B., Owen J., Bloor M.S., Hogg I., ***'Engineering document management strategy: analysis of requirements, choice of direction and system implementation'***, Proc Instn Mech Engrs, Vol, 211, Part B, 1997, p. 385-405.

[Hartley, 1992] Hartley J.R., ***'Concurrent Engineering: Shortening Lead Times, Raising Quality and Lowering Costs'***, Productivity Press, Cambridge, Massachusetts, 1992.

[Hee, Aalst, 2001] Hee K. van, Aalst W.P.M. van der, ***'Workflow management: models, methods and systems'***, BETA research institute, 2001.

[Helms, 2000] Helms R.W., '***PDM function model***', RapidPDM Project, deliverable D51, 2000.

[Hoevenaars, 1994] Hoevenaars S.D.W., '***Herontwerp van de logistieke besturing bij IGA***', Report Stan Ackermans Institute, Eindhoven University of Technology, 1994.

[Hutjes & Van Buuren, 1992] Hutjes J.M., Buuren J.A. van, ***'De gevalstudie: Strategie van kwalitatief onderzoek'***, Boom, Amsterdam, 1992.

[Katz, 1990] Katz R.H., ***'Towards a Unified Framework for Version Modeling in Engineering Databases'***, ,ACM Computer Surveys, Vol. 22, No. 4, 1990, p. 375-408.

[Korbijn, 1999] Korbijn A., ***'Vernieuwing in product ontwikkeling: strategie voor de toekomst'***, Stichting Toekomstbeeld der Techniek, Den Haag, 1999 (In Dutch).

[Krishnan, 1996] Krishnan V., ***'Managing the simultaneous execution of coupled phases in concurrent product development'***, IEEE transactions on Engineering Management, Vol. 43, No. 2, 1996, p.210-217.

[Krispijn, 2000] Krispijn G-J., '***Productivity analysis of home-office engineering for ABB Lummus Global***', graduation thesis TU Delft, 2000.

[Loch and Terwiesch, 1998] Loch C.H., Terwiesch C., '***Communication and Uncertainty in Concurrent Engineering***', Management Science, Vol. 44, No. 8, 1998.

[MacKrell, 1992] MacKrell J., ‘PDM Benefits Guide: ***Assessing Product Data Management Systems for Engineering and Manufacturing’***, CIMdata, Ann Arbor, Michigan, USA, 1992.

[March, 1994] March A., Usability: the new dimension of product design’, Harvard Business Review, 1994, p. 144-149.

[McIntosh, 1995] McIntosh K.G., ***‘Engineering Data Management: A Guide to Successful Implementation’***, McGraw-Hill Book Company, 1995.

[Miles & Huberman, 1984] Miles M., Huberman A., ***‘Qualitative data analysis’***, SAGE Publications, New York, 1984.

[Millson *et al.*, 1992] Millson M.R. *et al.*, ***‘A survey of major approaches for accelerating new product development’***, Journal of Product innovation Management, Vol. 9, 1992, p. 53-69.

[Muntslag, 1994] Muntslag D.R., '***Managing Customer Order Driven Engineering***', CIP-DATA Koninklijke Bibliotheek, Den Haag, 1993.

[NNI, 1999] Nederlands Normalisatie-instituut, ***‘Ontw. NEN-EN-9001’***, Nederlands Normalisatie-insituut, 1999 (In Dutch).

[Obank et al., 1995] Obank A., Leaney P., Roberts S., ‘***Data management within a manufacturing organization***’, Integrated Manufacturing Systems, vol. 6, no. 3, 1995.

[OMG, 1996] Object Management Group, '***Product Data Management Enablers***', Request for Proposal, OMG document: mfg/96-08-01.

[Pahl & Beitz, 1984] Pahl G., Beitz W., ***‘Engineering Design’***, Design Council, London, 1984.

[Pels, *et al*, 1995.] Pels H.J., Rexwinkel H.H., Hoevenaars S.D.W., ***‘Document quality levels: a PDM-tool for engineering-process control’***, Proceeding PDM-Europe ’95 Conference 24-26 October in Noordwijk, CIMdata, Ann-Arbor, USA, 1995.

[Peltonen *et al.*, 1996] Peltonen H., Pitkänen and Sulonen R., '***Process-based view on product data management***', Computers in Industry, Vol. 31, No. 3, 1996, p. 195-204.

[Reinertsen, 1997] Reinertsen D.G., ***‘Managing the Design Factory: a product developers toolkit’***, The Free Press, New York, 1997.

[Romney, 1995] Romney M., ‘***Business Process Reengineering***’, Internal Auditor, June 1995, page 24-29.

[Roozenburg & Eekels, 1995] Roozenburg N.F.M. and Eekels J., '***Product Design: Fundamentals and Methods***', John Wiley & Sons, 1995.

[Rumbaugh *et al.*, 1996] Rumbaugh J., Jacobson I., Booch G., ***'The unified modeling language reference manual'***, Addison-Wesley, 1996.

[Sabbagh, 1995] Sabbagh K., '***21st Century Jet: The making of the Boeing 777***', MacMillan, London, 1995.

[Sackett *et al.*, 1998] Sackett P.J. and Bryan M.G., '***Framework for the development of a PDM strategy***', International Journal of Production and Operations Management, Vol. 18, no. 2, 1998, p. 168-179.

[Schilling & Hill, 1998] Schilling M.A., Hill C.W.L., '***Managing the new product development process: Strategic imperatives***', Academy of Management Executive, vol. 12, no. 3, 1998.

[Simonse, 1998] Simonse L.W.L., '***Organisatie-ontwikkeling in Productcreatie***', dissertation Eindhoven University of Technology, 1998.

[Smith, 1997] Smith R.P., '***The Historical Roots of Concurrent Engineering Fundamentals***', IEEE Transactions on Engineering Management, bol. 44, no. 1, 1997.

[Stevens, 1993] Stevens, F., '***Vele eersten zullen de laatste zijn***' (in Dutch), Frits Philips Institute for Quality Management, Eindhoven, The Netherlands, 1993.

[Strauss & Corbin, 1990] Strauss A., Corbin J., ***'Basics of Qualitative Research'***, SAGE Publications, New York, 1990.

[Suh, 1990] Suh N.P., '***The Principles of Design***', Oxford University Press, 1990, ISBN 0-19-504345-6.

[Swanborn, 1996] Swanborn P.G., ***'Case-study's: Wat, wanneer en hoe?'***, Boom, Amsterdam, 1996 (In Dutch).

[Takeuchi & Nonaka, 1986] Takeuchi H., Nonaka I., '***The new product development game***', Harvard Business Review, January-February, p. 137-146, 1986.

[Tichem, 1997] Tichem M., ***'A Design Coordination Approach to Design For X'***, dissertation University of Delft, 1997.

[Ullrich & Eppinger, 1995] Ullrich K.T., Eppinger S.D., ***'Product design and development'***, McGraw Hill, London, 1995.

[Van Strien, 1986] Strien P.J., ***'Praktijk als wetenschap'***, Van Gorcum, Assen, 1986.

[Van den Kroonenberg & Siers, 1993] Kroonenberg H.H. van den, Siers F.J., 'Methodisch Ontwerpen', Educaboek, Culemborg, 1993 (In Dutch).

[Vroom, 1996] Vroom R.W., '***A general example model for automotive suppliers of the development process and its related information***', Computers in Industry, Elsevier Science, 1996, p. 262.

[Wallace, 1993] Wallace K., ***'A systematic approach to engineering design'***, In: Design management: a handbook of issues and methods (Oakley ed.) Cambridge, UK, 1993.

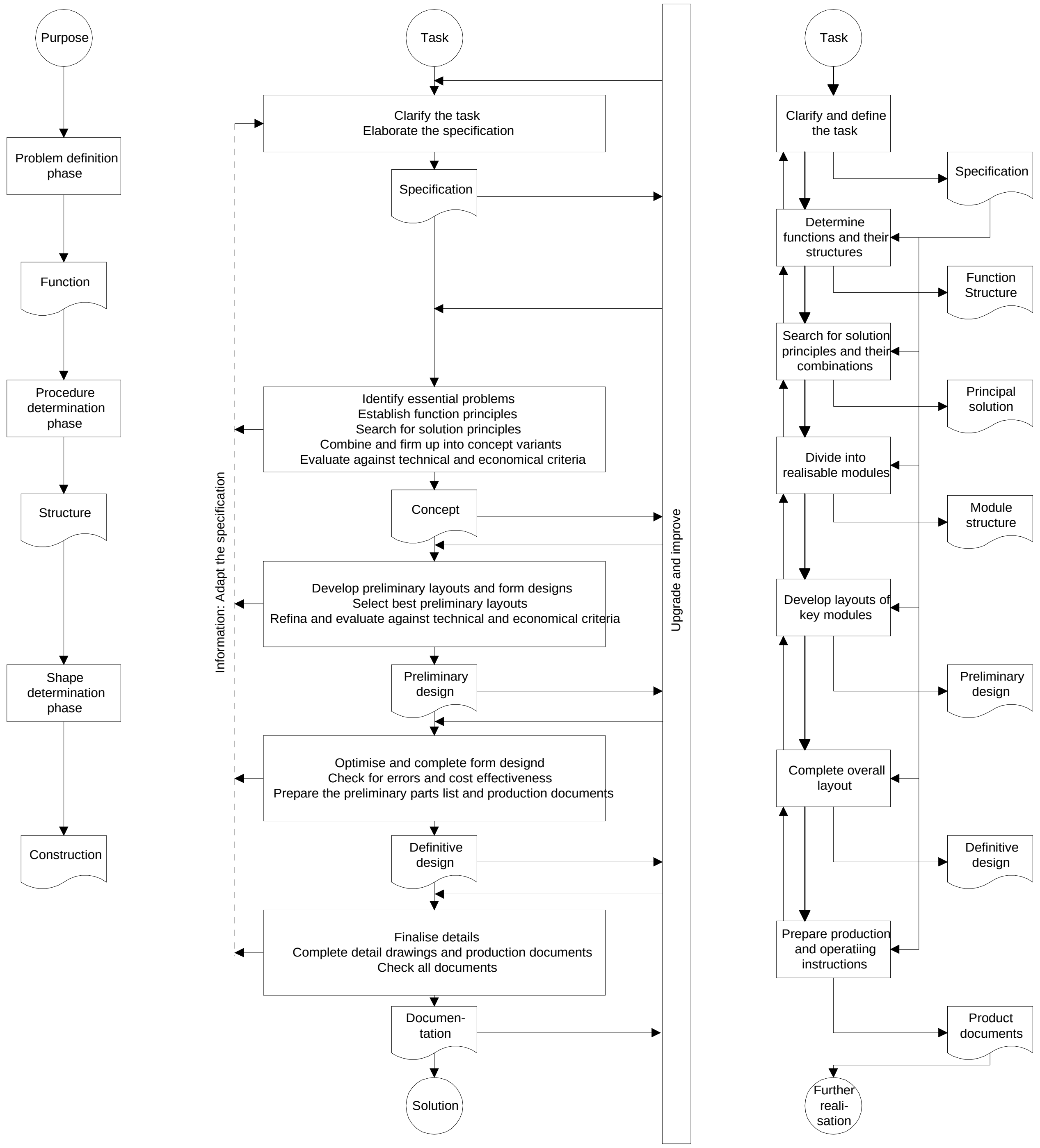
[Wheelwright & Clark, 1993] Clark K.B., Wheelwright S.C., '***Managing new product development - Text and Cases***', Harvard Business School, 1993

[Winner *et al.*, 1988] winner R.J., Pennel J.P., Bertrand H.E., Slusarczuk M.M., ***'The role of concurrent engineering in weapons systems acquisition'***, IDA R-338, Institure for Defense Analyses, 1988.

[Wortmann *et al*., 1997] Wortmann J.C., Muntslag D.R. en P.J.M. Timmermans, ***'Customer Driven Manufacturing'***, Chapman and Hall, 1997.

[Yin, 1994] Yin R.K., 'Case Study Research: Design and Methods', SAGE Publications, Thousands Oaks, USA, 1994.

Appendix - 1: Comparison of Descriptive Models



Appendix – 2: Details of the Aero case study

Project details

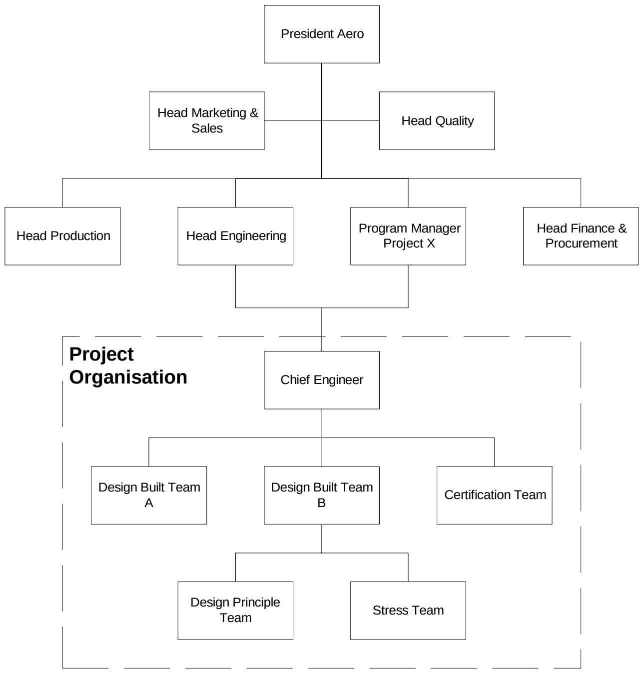
The project that is studied in this case study is the Pressure Bulkhead project. A Pressure Bulkhead is a part of an aircraft that is being developed by a large aircraft company. It concerns the part of the plane to which the landing gear is attached.

An aircraft integrator is responsible for the integration of the parts of the aircraft that are delivered by several suppliers and contractors. One of the suppliers is Aero or according to their own terminology they are acting as a main supplier. As a main supplier they take care of the development of the pressure bulkhead and its production process working to the customer's functional specifications.

The project started early in 1998 and during peak capacity 35 employees from Engineering where involved in the Pressure Bulkhead project. Apart from Engineering employees from Production, Program Support, and Purchasing have also been involved.

Aero already shipped the first ship set of the pressure bulkhead in August 1999 to the assembly plant of the system integrator. At this plant the pressure bulkhead is integrated into the first of the aircrafts. The first test flight of this new aircraft, however, is scheduled for January 2001 and certification is expected in the spring of 2002.

This is already an indication of the overlap of the processes in the development of an aircraft, the first aircrafts are already assembled before the test flight and certification has taken place. If the test flight or certification generates changes not only drawings need to be adjusted but also the aircrafts that have been assembled so far have to be changed. Obviously Time-to-Market is that important that it justifies such a degree of overlap.



Project organisation

Aero is a typical matrix organisation (see figure). They are organised in functional disciplines such as marketing & sales, engineering, production, and quality assurance. For the duration of a project or program they create a project team with members from these disciplines. These project teams are led by project or program managers while the functional disciplines are led by functional unit managers.

In this research the focus is on product development therefore only the project engineering organisation is discussed in more detail. A Chief Engineer leads the project engineering team and is ultimately accountable for the technical quality and integrity of the product. The project-engineering organisation consists of a Design Built Team and a Certification Team. The Design Built Team directs a Design Principle Team and a Stress Team.
Design Build Teams (DBT) are multi-disciplinary teams under the operational responsibility of a Chief Engineer. Disciplines that are part of a Design build team include Stress Engineering, Design Engineering, Materials/Processes and Technology Support and Production disciplines with respect to tooling and fabrication methods.
The Design Principle Team and the Stress Team are mono disciplinary teams who conduct the design engineering and stress engineering activities respectively. The Certification Team focus on the certification of the pressure bulkhead. Certification is an important aspect in the aerospace industry because it proves the airworthiness of a product.

All teams are located at the same geographical area. The Design Principle Team, Stress Team, Chief Engineer and DBT leader are located in the same office. The other DBT members are located elsewhere at the same location within 1000 meters.
The customer is not part of one of the teams and does not reside at the premises of Aero. However, at the start of the project there was an intensive co-operation with the customer at their premises. During this phase the scope of Aero's work package is precisely defined as well as the interfaces with other work packages. Once the scope and the interfaces were defined Aero could perform its activities relatively independent of the customer and other contractors at their own premises.

Project approach
For the pressure bulkhead Time-to-Market is one of the main business drivers. The aircraft company already made commitments to airline companies in regard to the delivery of new aircrafts. The delivery date is used as the starting point and from that day they calculate backwards to determine when the specific parts of the aircraft need to be available on the site of the system integrator. This is especially true for the first pressure bulkheads that have to be delivered to the system integrator. In order for the first pressure bulkhead to be delivered they have to do all the development work. As soon as the first pressure bulkhead is delivered the engineering team can focus on cost reduction which involves optimising the product design and the production process.

Besides Time-to-Market there are two other important issues in the aerospace industry. The first one was already mentioned and concerns safety. Under no circumstances it is allowed to jeopardise the safety of the aircraft. The use of cheaper materials is not allowed in any circumstances if it affects safety.
The second one concerns the weight. The lower the weight of the pressure bulkhead the lower the weight of the aircraft what results in a lower kerosene consumption. If the weight of the product exceeds a certain maximum Aero has to pay a significant penalty. On the other hand there are incentives for Aero to make the product lighter than the maximum weight.

Appendix 3 - Detailed description of EPC process at Chemical

1. Basic Engineering

The starting point for the Basic Engineering phase is the Basis of Feasibility (BoF) that is prepared by the customer. In this document the main requirements are listed with respect to the SMPO plant. During the Basic Engineering stage process designers and engineers are involved. They start with the preparation of the Basis of Design (BoD). The BoD contains the basis principles for the design and are the starting point for determining the processes that are required to produce the quantity and quality of the product specified in the BoF. The process is simulated using simulation software, which is based on Heat and Mass Balances (HMB). Simultaneously they also make Process Flow Diagrams (PFD) that schematically show the required equipment.

The PFD's are the main input for developing the most important documents for all consecutive activities: Piping & Instrumentation Diagrams (P&ID) or Process Engineering Flow Schemes (PEFS/UEFS). These documents schematically show all the equipment as well as the relations between the pieces of equipment in terms of piping, control loops and so on.

The Basic Engineering phase ends when the Process Data Sheets for equipment (*i.e.* vessels, compressors, and pumps) as well as the Equipment List are completed. The PEFS, Equipment List, and Datasheets are the major output of the Basic Engineering phase. In Chemical's case they are only doing Basic Engineering. This information is delivered in the form of a Basic Design and Engineering Package (BDEP).

2. Detailed Engineering

In this section the Detailed Engineering activities of the different engineering disciplines are described. The disciplines work concurrently on the detailed specifications of the plant.

Mechanical

The main input of Mechanical Engineering is the Process Data Sheets, which describe the process parameters and the required solutions, *i.e.* pieces of equipment. The job of the Mechanical department is to make a detailed specification of the required equipment. These specifications are sent to the vendors for enquiry, which is called a Requisition for Enquiry. The distribution of the requisitions goes via Seller Document Control (SDC) that takes care of copying and distributing the documents to the vendors.

It is the responsibility for the Procurement to collect the responses (*i.e.* bids) to the requisitions and to conduct the selection and negotiations. Mechanical is involved by Procurement to evaluate the technical aspects of the responses. If a vendor is selected Mechanical prepares the Purchase Order Requisition (POR) which is sent to the vendor via SDC and Procurement. From that moment Expediting and Logistics are responsible for getting the ordered equipment on site in time.

Both types of requisitions are distributed to other disciplines because they need information from the requisition. In the case of a vessel Civil needs information regarding the clamps that support their ladders and platforms while Piping needs information regarding the nozzle position to determine the hook-ups.

Piping
The main input for Piping is the PEFS these are used as input for the Piping Studies. The Piping Studies concern a first concept of the routing of the pipes (*i.e.* lines) that connect the pieces of equipment. These piping studies are produced manually on a drawing board
The Piping Studies are input for a group of engineers that compose the Overall Plotplan, which shows the arrangement of the major pieces of equipment. There are also Area/Section Plotplans that show the arrangement of equipment for an Area or Section in more detail.
The Piping Studies and the Plotplan results in a 3D model of the plant that is created in the PDMS system from CADCentre. For the details Piping needs information from the other disciplines regarding the in-line components, *i.e.* the equipment that is placed in the lines. An example of the information that is required is the nozzle orientation on vessels; this determines how a line is hooked-up to a vessel. All the details are also entered into the PDMS. The information in PDMS can be used to generate Isometrics[21]. These Isometrics are important for Construction because it tells them how to install and connect the lines in the plant.
The Isometrics that are generated from PDMS are managed using a system called PIAS. This system can generate the so-called Line List that contains a list of all the lines off the plant. Based on this list they can start the Material Take-Off (MTO), *i.e.* ordering the piping material. Similar items of the Line List are grouped. For each group of items they prepare a requisition that is send to the vendors via SDC and Procurement. Like the other disciplines they prepare a Requisition for Enquiry as well as the final Purchase Order Requisition.

Civil/structural
The main input for Civil/Structural is the Plotplan and the details of the equipment (*i.e.* Mechanical Datasheets) that need to be supported by concrete or steel structures. The Plotplan shows the location of the pieces of equipment while Mechanical Datasheets provide information about, for example, the weight and length of a vessel. The weight of the equipment determines the required foundation while the weight and position of the equipment requires the amount and position of piles (this is documented in the Piling Plan). The type, length and diameter of piles is determined by the condition of the soil. Therefore before they start they take samples of the soil that are studied in a lab (by a third party).
Another example of information that is required by Civil is information from Piping. It concerns information regarding the need for piping racks that support the pipes/lines. The Civil department needs this information because they design these steel structures, *i.e.* piping racks. Other things that are designed by Civil include fences, buildings, paves, and sewers.
Civil also takes care of all the requisitions (Requisitions for Enquiry as well as Purchase Order Requisitions) regarding the piles, concrete, sewers, buildings, and steel structures. Like the other departments they hand over the requisitions to SDC and Procurement for further processing.

Instrumentation
The main input for Instrumentation is the Process Control Philosophy and PEFS. Based on this information they determine the System Concept that describes how to control the plant. This is worked out in more detail in Instrument System Block Diagrams and Process Control Diagrams. This finally results in an important deliverable called the Instrument Index that lists all the required instruments.

[21] In the early days engineers had to make these Isometrics manually.

All instruments are connected to the DCS system that gathers process information such as temperatures, pressures, or flow rates. This means that data cables are required to transport this data from the instruments to the DCS system and vice versa. Based on the Plotplan and PEFS Instrumentation performs studies to define the (data) cable routings, *i.e.* layout drawings.
Principally all instrument data is stored in a system called INTOOLS. With this application it is possible to define dynamic links between for instance objects in a loop drawing and objects in a database so that both are always up-to-date and consistent. Another application that they use is PDMS. Piping and Civil are among the first that fill the 3D model in PDMS. But before Piping and Civil have completed their design Instrumentation will already put their instruments into the 3D model. Apart from junction boxes and actuators it also includes underground cable trenches. In other words they work on the 3D model simultaneously with the other disciplines.
After defining the required control system and instruments they have to prepare the requisitions (Requisitions for Enquiry as well as Purchase Order Requisitions). Like the other departments they hand over the requisitions to SDC and Procurement for further processing.

Electrical

The main input for Electrical is the Process Data Sheets, which in a later stage are replaced by the Mechanical Data Sheets. Firstly, they need this information because they have to make the specification for equipment such as electrical motors (low voltage and high voltage) and back-up power units. Secondly, it provides information on the power consumption of the different pieces of equipment. The power consumption determines what power cables, switchboards and so on are needed on the plant. The processing of this information will lead to the first key document called the Key Single Line Diagram (KSLD). It concerns a schematic overview of the high voltage/low voltage distribution. Later the information on the KSLD is further detailed which results in the Cable List, which is an overview of all the required cabling on the plant. For every cable(tree) the routing is determined which results in a large number of layout drawings (AutoCAD), in total approximately 1,500. The cable layout drawings are translated into detailed ‘work instructions’ for Construction, so-called Cable Real Schedules and Cable Real Drums. These ‘work instructions’ are generated using an Access application.
Electrical also prepares requisitions (Requisitions for Enquiry as well as Purchase Order Requisitions) for the required material and equipment. Like the other departments they hand over the requisitions to SDC and Procurement for further processing.

3. Procurement

In this section the activities of Procurement are discussed, this also includes activities of Expediting and Logistics.

Typical characteristics

The main responsibility of Procurement, Expediting and Logistics is to make sure that the required material and equipment is available on-site in time. Due to the tight schedule this is a complex job because one has to realise that it involves a lot of custom-made material and equipment. This means that the goods are not available at the moment they are ordered. Due to the long lead-times engineering changes can have a severe impact on the schedule as well as the cost. Moreover, it also involves some exceptionally large pieces of equipment, such as the towers, of more than 70 meters. Such items are transported to Asia by boat, therefore it takes up to several weeks before they arrive on the plant.

Purchasing process

A large part of the equipment is custom-made and is produced by third parties. Besides production services some of the third parties also provide engineering services. Chemical has to provide the specifications so that the third parties are able to manufacture the equipment. But before a Purchase Order is placed they send out Enquiry Requisitions to several vendors with the request to make an offer.

Such a requisition contains technical details (*e.g.* datasheets, drawing, and references to standards) defined by Engineering and commercial details defined by Procurement. The distribution of the requisitions is done by Seller Document Control (SDC). They check if the requisition is complete, make the requested number of copies and distribute the requisition to the vendors as well as to internal departments for information. SDC files all the requisitions for later reference so that there is an overview, which requisitions are sent out to the vendors.

The responses of the vendors (*i.e.* bids) are received by SDC and distributed to Procurement and Engineering. They also file a copy of the vendor's bid so that there is an overview of, which requisitions are completed and which are still pending. Engineering performs a technical bid evaluation and Procurement performs a commercial bid evaluation. This forms the input for the final vendor selection.

Placing the order means that a Purchase Order Requisition (POR) is send to the vendor. It contains an update of the technical specification because the Requisition Enquiry contained preliminary information. This POR is distributed and filed by SDC that also receives the final bid from the vendor, which is distributed by SDC to Procurement and Engineering.

Follow-up and transportation

Expediting is the department that monitors the actual delivery of material, equipment and components. Firstly, they follow up the Enquiry Requisitions that are send to the vendors. If there is no response from the vendor or if the bid is incomplete they take appropriate action. Secondly, they follow up the Purchase Orders that are sent to the vendors. For example they monitor if the vendors send a notice for the receipt of the order, if they shipped the items on time, and if the items arrived on-site on time.

The actual transportation is taken care of by the Logistics department that subcontracts this work to specialised freight carriers. These carriers collect the material and equipment at the vendor and transport it to the construction site in Asia.

Conflicting priorities

There are conflicting priorities between Procurement and Engineering. In order to get discounts Procurement wants to order all equipment of the same type at the same time. Hence a requisition that is written covers the specifications for a group of items of the same type, for example all control valves of a specific diameter. It is also possible that it covers the specifications for several groups of items that can be ordered from the same vendor.

However, Engineering is working its way through the plant per area that follows the Construction Schedule. In this way they can release the drawings and specifications per area so that construction can also work its way through the plant per area. However, Engineering will not have the specifications for a group of items available before they have worked their way through the complete plant. For Engineering it would make more sense to send separated requisitions for every area, however, than the purchasing cost are higher. Consequently Engineering has to make an estimate of the number of items in a group, for example the number of control valves. For this group they make a preliminary specification, based on the information that is available at that time, which is sent with the Enquiry Requisition to the vendors. The release of all the indexes and lists, created by the engineering disciplines (see previous section), is an important aid for estimating the required amount of equipment. Procurement needs to receive updates of these indexes and lists to see if quantities have changed or if equipment is to be replaced or stopped completely.

5. Construction

Engineering has to produce all the drawings and specifications of the plant to enable Construction to built the plant. In a Phased EPC project this information is provided to Construction in one batch. In a Concurrent EPC project this information has to be delivered in batches. This means that Engineering should first deliver those drawings and specifications that are needed first by Construction. This means that the Construction schedule drives the Engineering schedule.

The Construction of a plant is quite straightforward because there are limited alternatives to how it can be erected. It starts with the foundation that includes piling and concrete structures. This is often an important milestone because the foundation also includes underground piping, cable trenches and sewers. As soon as the foundation is completed it is impossible to add a cable trench. It is possible to put spare cables in the ground to cover contingencies. This might be an option for data cables, which are relatively cheap, but is not an option for power cables that are quite expensive.

When the foundation is complete the structures have to be put in place, which support the various pieces of equipment and piping as well as the required buildings such as a Control Room for instance. Finally the equipment can be put in place which is connected by the lines that contain in-line equipment such as valves, orifices, transmitters and so on.

It is not possible to put equipment in place if the structures, foundation, and piling are not in place. In other words there is a rather fixed order in which a plant has to be build. This order also determines the delivery of information from Engineering to Construction.

The Construction priorities, however, are in conflict with those of Engineering. From the start Construction needs detailed information on the location, orientation, length, and weight of equipment. Engineering on the other hand starts with studies before it gradually details the specification of the plant. Theoretically this means that Construction, for example, cannot start with piling before Engineering is completed because that is the moment that the details are available concerning the location and weight of equipment that is required as input for making the piling plan for example. The civil department, which creates the piling plan hence needs preliminary information from the other disciplines in order to complete the piling plan prior to Engineering finishing.

Appendix 4 - Examples of preliminary release at Chemical

Example 1: Instrumentation

The Instrumentation discipline is amongst others responsible for making the In-line Instrument Index. This is in fact a bill of material of all in-line instruments, *e.g.* valves, orifices, and transmitters. The index is derived from the Instrument Index, containing all instrument and not only in-line instruments, and the Instrument Data Sheets.

The Piping discipline needs the In-Line Instrument Index for making the Isometrics. In order to start as early as possible the Piping discipline works with preliminary release, *i.e.* AFD, of the In-Line instrument Index. This In-Line Instrument Index shows all the in-line instruments but their specifications are not precisely known yet. This is not a problem for the Piping disciplines to start incorporating the in-line instruments in their Isometrics.

If the supplier is selected Instrumentation it receives exact specifications of the in-line instruments. Based on this information Instrumentation updates the In-Line Instrument Index and releases it for AFC. The Piping discipline uses the AFC version of the In-Line Instrument Index to update their Isometrics. By comparing the AFD and AFC version they can see what is changed and hence what needs to be updated in their Isometrics.

Note: Apart from the In-Line Instrument Index there are also other indexes and lists, *e.g.* the Line List. For these indexes similar working practices are used.

Example 2: Civil

The Civil discipline is amongst other things responsible for the design of the steel structures. These steel structures are directly modelled in 3D in PDMS. The steel structures are made by sub-contractors that also do the detailed design of the steel structures.

To finish as early as possible the 3D models are released in two steps to the contractors. Firstly, Chemical sends an AFD model that is later followed by an AFC model. Based on the AFD model the sub contractor can start its detailed design activities, order material, and plan the production of the structure in its shopfloor. However, the sub contractor is not allowed to actually make the steel structure before they have received the AFC model from Chemical.

In case there are any changes after the model is released AFD, the Civil discipline can decide to send a Revised AFD to the sub contractor. If it only concerns minor changes they will not release a Revised AFD model and hence the sub contractor will not see these changes before it receives the AFC model.

The design of all steel structures is not finished simultaneously. Moreover, the steel structures are not needed on site simultaneously. Therefore, the Civil discipline releases the 3D models in small batches to the sub contractor. To manage this process they maintain a spreadsheet that contains a list of all the steel structures that are required. In this spreadsheet the sub contractor can find the planned AFD and AFC dates of the 3D models for each steel structure. Moreover, the spreadsheet also contains information about when the sub contractor should deliver its drawings and the steel structure to Chemical.

In other words by means of this spreadsheet Chemical has an overview of the progress of Chemical and the sub contractor. The spreadsheet is updated every two weeks and is also provided to the sub contractor.

Example 3: Piping

The Piping discipline is amongst other things responsible for providing the ISO metrics to Construction. These Isometrics are not provided as preliminary information to site, but they are released in batches. Planning and monitoring the Isometrics is of interest in this example.

Isometrics are drawings that present the lines of the plant in an isometric view. In the project more than 7000 of these drawings are made. The creation of the Isometrics starts with planning the amount of Isometrics that are required.
Planning is done on the basis of the Line List or Index. The Line Index is in fact a bill of material of all the lines that are needed in the plant. This Line Index is entered into a system called PIAS and for every line one or more Isometrics are needed. Therefore, several sheets (including document number) are planned for each line number in PIAS. Moreover, it is planned when these Isometrics should be available on site.
The Isometrics are not created from scratch and the 3D PDMS model is used to generate them. Piping engineers add additional information to the Isometrics that are generated. Isometrics also need to be approved formally before they are released. This complete lifecycle of the Isometric, from creation to release, is monitored using the PIAS system. Formerly monitoring the Isometric lifecycle was done completely manually.
With PIAS the Piping discipline is able to generate reports that show the status of a set of Isometrics. If it seems that some Isometrics are late they can take appropriate action. These status reports are also published on the Intranet. This enables Construction (located in Asia) to see which Isometrics will arrive in the coming weeks.

Appendix 5 - RapidPDM project

RapidPDM project
The RapidPDM-project, an Esprit-project, started in September 1998 to develop methods and tools that help to reduce the time and cost needed to implement a PDM system and involves every phase of the lifecycle of a PDM implementation[22]. The RapidPDM project is described here because part of this research is conducted within the framework of the RapidPDM project.

Industrial Problem
PDM vendors, CAD Vendors and independent Systems Integrators now offer the market an expanding range of electronic PDM systems. Implementation of a PDM system in an organisation takes considerable time and effort for customisation and configuration of the software and Business Process Redesign. This effort often exceeds the purchase cost of the software by a factor of 3 or more. These high implementation costs create a significant barrier for enterprises, especially Small and Medium-size Enterprises (SMEs), considering an investment in PDM. The implementation process must be better understood, structured in manageable steps and supported with proper tools to reduce the resources required for successful implementation and to make Product Data Management a viable option for SMEs.

Objectives
The primary objective of the RapidPDM project was to develop a generic PDM implementation methodology that is supported with a set of IT-tools. This addresses the needs of Industry by supporting the implementation process for Product Data Management. The objectives of this project were to:

- develop methods to structure, manage and speed up the implementation process,
- develop tools to support the implementation process, also in SMEs,
- make the methodology available to manufacturing industry through Workbooks, Courses and educational materials,
- make the tools available to PDM-vendors and consultants so that they can be used in the market.

Results
The result of the RapidPDM project include a tested methodology and a prototype workbench that can be used by PDM consultants to enable manufacturing companies to implement PDM in a shorter time and at a lower cost. The methodology is documented and published on paper and in a digital format. A course is developed to instruct consultants on how to apply the methodology and to use the tools. Validation is executed as part of the project. The software vendor partner plans to develop the prototype into a commercial product. The consultant partners plan to market the course in different European countries in order to educate a body of trained PDM-consultants. The academic partners plan to make the course available in their curricula.

See also http://www.rapidpdm.org

[22] More information about the RapidPDM project can be found at: http://www.rapidpdm.org

7.6.1 Link with this thesis

There are two results of the RapidPDM project that are used in this thesis:

- PDM Function model: The function model describes the nine main functions of a PDM system. This function model makes it possible to speak about PDM in terms of individual functions.
- Conceptual simulation model: This model is used as a research model in this research. It describes how a concurrent product development process can be modelled from a document point of view. This model is used to explain how product development activities start preliminary information.

Summary (English)

Reason for this research

Changes in competition, customer demands, and technology are the forces that have created a competitive imperative for speed, efficiency and high quality in the development process. This research focuses on one aspect of improving product development processes: reducing cycle time. This can be achieved in several ways, amongst others by redesigning business processes. One of the most well known concepts in this context is Concurrent Engineering. It focuses on reducing cycle time by overlapping phases of the product development process, also known as *lifecycle integration*.

Concurrent Engineering requires frequent, face-to-face, bilateral communication of preliminary information instead of late release of complete information (which is the case in a sequential process). Companies apply multidisciplinary teams to facilitate close co-operation and the exchange of preliminary information. There has been a lot of research with respect to these disciplinary teams. But there has been considerably less research on how to manage incomplete or preliminary information within these multidisciplinary teams.

The goal of this research is to answer the question how starting on preliminary information can be supported by ICT. The focus is on so-called Product Data Management (PDM) systems that manage the product definition data across the whole lifecycle of the product. In case ICT is used to support business processes often the term enabler is used. This term stems from the theory on Business Process Re-design of the early 90's. The term indicates that ICT offers new opportunities for the (re-)design of business processes, which should lead to significant performance improvements. The research question therefore is *how* Product Data Management functions can act as an enabler for starting on preliminary information and under what conditions these functions act as an enabler.

Research design

The objective of this research is to develop a theory that describes which PDM functions act as an enabler for starting on preliminary information and under which conditions. This theory is based on literature research in the field of Concurrent Engineering, Product Data Management, and Business Process Re-design. The literature research has been used to develop a research model to better understand the research problem. The model has been used to formulate a number of predictions on starting on preliminary information and the use of PDM functionality. These predictions have been validated in two case studies in order to support the developed theory from which the predictions are derived.

The knowledge of two experts has been used to select the case study companies. They were asked to identify (product development) companies in the Benelux that start on preliminary information supported by PDM functionality. For the final selection of two case study companies several independent and dependent variables were taken into account. The independent variables are supposed to have no effect on the outcome of the research. It involves variables such as the industry in which the companies operate, the type of product and the production method they use. The two companies that have been selected differ on all three variables to validate this assumption. On the other hand the dependent variables are supposed to have an effect on the outcome of the research. It involves the product complexity, product modularity, number of changes, and the geographical spread of employees. The two companies that have been selected are similar on all four variables in order to be able to compare the results.

Starting on preliminary information

This research focuses on starting on preliminary information in product development processes. The output of the product development process is product definition information, for example calculations, drawings or reports, which is required by production in order to be able to produce the product. Product definition information is still mainly exchanged in the form of documents (either analogue or digital). Product definition information is produced by a network of activities in the product development process, i.e. the output of one activity is used by another activity as input. In a traditional product development process activity B cannot start before activity A released the documents that activity B needs as input. Therefore activity B will not start before activity A is completed, hence resulting in a sequential process. In case activity A and B are conducted concurrently, activity B starts before activity A is completed. This overlap is realised by exchanging preliminary document versions between activity A and B. In that case activity B already starts on basis of preliminary document versions that are released by activity A. The advantage of starting on preliminary information is that activity A and B can be completed in a shorter timeframe.

Consequences of starting on preliminary information

An important characteristic of starting on preliminary information is that smaller batches of documents are released one or more times in preliminary form before their final release. This is done to create overlap between activities in the product development process and requires much more co-ordination and planning than releasing document versions in one large batch in their final form. Starting on preliminary information requires that the progress of small sets of documents is monitored to make sure that preliminary document versions are released according plan.

Besides there is a risk that the content of preliminary released document versions changes. If this is the case the outdated information should be replaced by the new information as soon as possible because using the outdated information results in higher cost and delays. Therefore it is important that the impact of a change is determined as quickly as possible so that it is known which documents are affected by the change. This could be done using the product structure and the documents that are related to parts of this product structure. Once it is known which documents are affected it should be determined if versions of these documents have already been preliminarily released. The people that received a preliminary version of a document that is affected by the change should be notified as quickly as possible.

Starting on preliminary information in a controlled manner requires a lot of administration that needs to be updated continuously because of the changes. Therefore it is assumed that PDM functionality (in a commercial off-the-shelf system or in-house developed system) can act as an enabler in this context. The application of PDM functionality can reduce the administrative effort that is caused by starting on preliminary information. Moreover, it is expected that the use of PDM functionality increases the transparency of the product development process. But it is not expected that the application of PDM functionality make sense in all situations. In small projects, for example, in which only a few people are involved it is expected that starting on preliminary information can be organised and co-ordinated informally. But in complex projects in which dozens of people from different disciplines and on different locations are involved this can no longer be organised and co-ordinated informally. In other words the complexity of the project determines if PDM acts as an enabler for starting on preliminary information or not.

Case study results

Both case study companies start on preliminary information by releasing preliminary document versions. The quality of the document is expressed in the status of the document that is also used as an indicator for measuring progress on document as well as project level. In both cases the preliminary release of document versions is planned beforehand and monitored by registering the progress of each individual document. One of the two case study companies uses a conventional method towards starting on preliminary information and uses mainly spreadsheets to control this process. Because of the relative low complexity of the project there is no direct need to use more sophisticated control systems. In the other case study company the project complexity is higher therefore they have developed systems (in-house) that contain PDM functionality to control this process. This system makes it much easier for them to generate status and progress reports, moreover they have instant access to the status of projects.

None of the two case study companies registers the product structure and the related documents in order to support the change impact analysis, because of the administrative effort that is involved. Besides the case study results show that determining the documents that are affected by a change is only a small part of the total change process. The advantage of only supporting this part of the process is therefore relatively small. However, both companies indicate that they would reconsider registering the product structure and the related documents if they would have the means to do this more efficiently.

To identify who received a copy of a preliminary document version both companies use distribution lists. There is a distribution list for each document type, it defines that each instance of that document type should be send to a number of discipline leaders that are on the list. This is a very coarse distribution list because and companies are not able to define and maintain distribution lists for each individual document version because of the administrative effort. In this context a PDM system could act as an enabler because it can easily maintain a distribution list for each specific document version. Besides it also reduces the manual effort of distributing the released document versions by doing it electronically instead of on paper. Finally the case studies showed that PDM functionality does not necessarily result in faster notification of changes. Often people are already informed informally during a meeting or by a telephone call before they receive the formal change notification via the PDM system.

Conclusions

The research resulted in a theory on starting on preliminary document versions. The companies included in this research start on preliminary document versions as predicted by this theory. This was illustrated using a method for describing the preliminary release process that has been developed in this research. The research also shows that the administrative effort increases, as a result of more planning and co-ordination, and that both companies therefore use some kind of PDM functionality. The use of this PDM functionality results in the predicted performance improvements of the product development process. Besides efficiency improvements also the transparency of the product development process increases making it easier to control this process.

However, the researched companies did not use all the PDM functionality as predicted beforehand. Therefore they also did not realise all the predicted performance improvements. This led to the conclusion that the *maturity level* of the company influences the PDM functionality that is used and to what extent it is used. The term maturity stems from the literature on Capability Maturity Models in which several levels of organisational maturity are distinguished. The maturity level of an organisation determines the level of ICT support that is required. In this research the predictions with respect to used PDM functionality were based on the highest level of organisational maturity. From this can be concluded that both case study companies are not on this highest maturity level, because they did not use all the PDM functionality that belongs to the highest level of organisational maturity.

Summary (Dutch)

Aanleiding

Door veranderingen in de concurrentie, de vraag van de markt en de technologie is de nadruk steeds meer te komen liggen op een snel, efficiënt en kwalitatief hoogwaardig productontwikkelproces. Dit onderzoek richt zich op één aspect van het verbeteren van het productontwikkelproces namelijk het verkorten van de doorlooptijd. Dit kan op meerdere manieren worden bereikt, waaronder het herinrichten van de productontwikkelprocessen. Eén van de bekendste methoden is Concurrent Engineering waarbij de doorlooptijd wordt verkort door het overlappen van fasen in het productontwikkelproces, ook wel *lifecycle integration* genoemd. Concurrent Engineering vereist het frequent, face-to-face, bilateraal uitwisselen van voorlopige informatie in plaats van het laat vrijgeven van definitieve informatie. Om dit te bewerkstelligen worden multidisciplinaire teams toegepast in het productontwikkelproces. Er is veel onderzoek gedaan op het gebied van deze multidisciplinaire teams, er is echter veel minder onderzoek gedaan naar hoe deze teams voorlopige informatie met elkaar uitwisselen.
Dit onderzoek richt zich op de vraag hoe het starten op voorlopige informatie kan worden ondersteunt door middel van ICT. Er wordt daarbij specifiek gekeken naar zogenaamde Product Data Management systemen, deze systemen beheren de product definitie gegevens gedurende de gehele levenscyclus van een product. Wanneer bedrijfsprocessen worden ondersteunt met ICT wordt vaak de term enabler gebruikt. Dit is een term die stamt uit de theorie over Business Process Redesign van de begin jaren ’90. Het geeft aan dat ICT nieuwe mogelijkheden biedt voor het (her)inrichten van de bedrijfsprocessen die tot significante performance verbeteringen kunnen leiden. In dit onderzoek is de vraag *hoe* Product Data Management functionaliteit als enabler kan fungeren voor het starten op voorlopige informatie en onder welke condities.

Opzet van het onderzoek

Het doel van het onderzoek is te komen tot een theorie die beschrijft welke PDM functionaliteit als enabler fungeert voor het starten op voorinformatie en onder welke condities dit het geval is. Deze theorie is vooral gebaseerd op literatuuronderzoek op het gebied van Concurrent Engineering, Product Data Management en Business Process Redesign. Op basis hiervan is een onderzoeksmodel opgesteld om de problematiek beter te kunnen doorgronden. Dit heeft geresulteerd in een aantal voorspellingen over het starten op voorinformatie en het gebruik van PDM functionaliteit daarbij. Deze voorspellingen zijn in een tweetal case studies getoetst met als doel de bovenliggende theorie te valideren.
Voor de selectie van de case studies is de hulp ingeschakeld van een aantal experts. Hen is gevraagd aan te geven welke (productontwikkel) bedrijven in de Benelux starten op voorinformatie en daarbij gebruik maken van PDM functionaliteit. Bij de selectie van de twee bedrijven is rekening gehouden met een aantal variabelen wel/geen rol spelen. Allereerst zijn er een aantal onafhankelijke variabelen vastgesteld waarvan wordt verwacht dat ze geen invloed hebben op de resultaten. Dit zijn de industrie waarin het bedrijf opereert, het soort product dat ze maken en de gehanteerde productie methode. Omdat verwacht wordt dat ze geen invloed hebben zijn twee bedrijven geselecteerd die op al deze drie variabelen van elkaar verschillen. Daarnaast zijn er ook een aantal afhankelijke variabelen vastgesteld. Dit zijn de product complexiteit, product modulariteit, aantal wijzigingen in het ontwerp en de geografische spreiding van de medewerkers. De bedrijven die zijn geselecteerd dienen zoveel mogelijk gelijkt te scoren op deze variabelen zodat de resultaten vergelijkbaar zijn. Dit heeft uiteindelijk geresulteerd in de selectie van een tweetal bedrijven waar de van de theorie afgeleide stellingen zijn getoetst.

Starten op voorlopige informatie

Het onderzoek richt zich op het starten op voorinformatie in het productontwikkelproces. De output van dit proces is product definitie informatie, bijvoorbeeld in de vorm van berekeningen en tekeningen, welke nodig is om het product te produceren. Product definitie informatie wordt veelal nog steeds vastgelegd in de vorm van documenten welke worden geproduceerd door een netwerk van activiteiten waarbij de ene activiteit de output van een andere activiteit als input nodig heeft. In een traditioneel productontwikkelproces kan een activiteit B pas starten als de documenten van activiteit A zijn vrijgegeven. Activiteit B start dus niet voordat activiteit A in zijn geheel is afgerond. Wanneer activiteit A en B overlappend worden uitgevoerd, start activiteit B al voordat activiteit A in zijn geheel is afgerond. Deze overlap wordt gerealiseerd door tussentijds voorlopige versies van documenten tussen activiteit A en B uit te wisselen waarmee activiteit B alvast kan starten. Het voordeel is dat dezelfde activiteiten in een korter tijdbestek kunnen worden uitgevoerd.

Gevolgen van starten op voorlopige informatie

Een belangrijk kenmerk van het starten op voorlopige document versies is dat kleinere batches documenten, een of meerdere malen *voorlopig* worden vrijgegeven alvorens ze definitief worden vrijgegeven (om zoveel mogelijk overlap te creëren). Dit vereist veel meer planning en coördinatie dan het in één grote batch vrijgeven van alleen definitieve document versies. Dit betekent dat de voortgang van documenten nauwkeurig dient te worden gevolgd om ervoor te zorgen dat alles volgens planning (voorlopig) wordt vrijgegeven.

Daarnaast is er het risico dat voorlopig vrijgegeven document versies nog wijzigen. Er dient te worden voorkomen dat in geval van een wijziging te lang met verouderde document versies wordt doorgewerkt. Hoe langer met verouderde document versies wordt doorgewerkt, hoe hoger de kosten en langer de vertraging. Hiervoor is het van belang dat zo snel mogelijk de gevolgen van een wijziging worden bepaald, zodat bekend is in welke documenten een wijziging optreed. Door de product structuur en de daaraan gerelateerde documenten te registreren kan worden bepaald welke documenten door een wijziging wordt beïnvloedt (impact analyse). Vervolgens dient bekend te zijn of deze documenten al in voorlopige vorm zijn vrijgegeven en aan wie ze dan zijn vrijgegeven. Deze personen dienen dan zo snel mogelijk van de wijziging op de hoogte te worden gebracht.

Vanwege de hoeveelheid informatie die hiervoor dient te worden geregistreerd en het aantal mutaties in deze informatie is de stelling dat PDM functionaliteit (in een commercieel PDM systeem of een zelf ontwikkelde applicatie) hierin als enabler kan fungeren. De PDM functionaliteit kan de hoeveelheid administratieve lasten reduceren en de transparantie van het proces verhogen. Echter het is de verwachting dat PDM functionaliteit niet in alle gevallen zinvol zal zijn. In kleine projecten bijvoorbeeld, waarin slechts enkele personen zijn betrokken, kunnen wijzigingen op een informele wijze worden afgestemd. Echter bij meer complexe projecten, waarin tientallen personen van verschillende disciplines van verschillende locaties betrokken zijn, is dit niet meer mogelijk. De complexiteit van projecten is dus mede bepalend voor het feit of PDM functionaliteit een enabler is.

De resultaten van de case studies

Beide bedrijven starten op voorlopige informatie door het vrijgeven van voorlopige document versies. De kwaliteit van het document wordt daarbij uitgedrukt in de document status die wordt gebruikt om de voortgang van het document en derhalve het proces te meten. De (voorlopige) document vrijgaven worden in beide bedrijven gepland en gevolgd, waarbij de voortgang van iedere individuele document versie wordt geregistreerd. Een van de twee bedrijven doet dit op conventionele wijze in een spreadsheet, gezien de relatief lage

complexiteit van het project is dit geen probleem. In het andere bedrijf is de complexiteit hoger en heeft men een eigen systeem ontwikkeld met PDM functionaliteit. Hierdoor is het veel eenvoudiger om voortgang en status rapportages te genereren, bovendien heeft men altijd een actueel inzicht in de voortgang van het project.
Geen van de case study bedrijven registreert de product structuur en de relatie met documenten ten behoeve van de impact analyse, met name vanwege de benodigde administratie. Daarnaast laat het onderzoek zien dat het bepalen van welke documenten door een wijziging worden beïnvloed (impact analyse) maar een klein onderdeel is van het proces waarin de gevolgen van een wijziging worden bepaald. Het voordeel van het ondersteunen van alleen dit deel van de impact analyse is dus maar gering. De case study bedrijven geven echter wel aan dat men zou overwegen om de product structuur en de relatie met documenten te registreren wanneer men een instrument zou hebben om dit efficiënt te kunnen doen.
Beide case study bedrijven registreren, tot op zeker hoogte, wie een voorlopige versie van een document ontvangt door middel van distributielijsten. Per document type is een distributielijst opgesteld die naar een aanspreekpunt per discipline wordt gestuurd. Echter men is niet in staat om van ieder individuele document versie vast te leggen waar het zich op een bepaald moment bevindt omdat dit te veel administratie met zich mee zou brengen. Hier zou een PDM systeem als enabler kunnen optreden doordat een PDM systeem dit eenvoudig kan registreren. Voor elk individueel document kan een specifieke distributielijst worden bijgehouden. Daarnaast kan de administratieve last van het handmatig distribueren van (voorlopig) vrijgegeven informatie enorm worden gereduceerd. Voor de snelheid waarmee een persoon wordt ingelicht omtrent een wijziging blijkt een PDM systeem niet essentieel. Vaak zijn personen informeel al ingelicht, in een vergadering of per telefoon, voordat het formele bericht bij de persoon arriveert.

De conclusie van het onderzoek
Het onderzoek heeft geresulteerd in een theorie over het starten op voorlopig vrijgegeven document versies en de onderzochte bedrijven bleken inderdaad volgens de voorspelde wijze te werken. Hierbij is gebruik gemaakt van de ontwikkelde methode voor het beschrijven van processen voor het starten op voorlopige informatie. Het onderzoek laat tevens zien dat de administratieve lasten toenemen als gevolg van het starten op voorlopig vrijgegeven document versies en dat bedrijven daarom PDM functionaliteit gebruiken. Het gebruik van deze PDM functionaliteit leidt tot de voorspelde performance verbetering van het proces. Het betreft efficiency verbeteringen maar belangrijker is dat transparantie van het proces is toegenomen waardoor het beter kan worden beheerst.
Echter de onderzochte bedrijven bleken niet alle voorspelde PDM functionaliteit te gebruiken waardoor ook niet alle verwachte performance verbeteringen zijn gerealiseerd. Dit heeft geleid tot de conclusie dat het *maturity level* van een organisatie bepalend is voor het feit welke functionaliteit wordt gebruikt. De term maturity komt uit de theorie over Capability Maturity Models waarin verschillende niveaus van organisatie ontwikkeling worden onderscheiden, de maturity levels. Afhankelijk van het maturity level kan een bepaalde mate van ICT ondersteuning worden gerealiseerd. In het onderzoek is uitgegaan van de PDM functionaliteit die hoort bij het hoogste niveau van maturity. Derhalve dient te worden geconcludeerd dat de twee onderzochte bedrijven zich niet op dit niveau bevonden en derhalve ook niet alle daarbij behorende functionaliteit gebruikten.

Curriculum Vitae

Remko Helms was born on October 11, 1970 in Amerongen. In 1989 he received his VWO diploma from the Amerfoorste Berg in Amersfoort, after which he started studying Industrial Automation at the Hogeschool Eindhoven. He received his degree, cum laude, in 1993 and started studying Industrial Engineering and Management Sciences at the Eindhoven University of Technology. In 1996 he received his degree, cum laude, after a research project at BW/IP International in Roosendaal concerning the cost and benefits of Product Data Management (PDM).

During this research project he met Arjan de Kok from MiQ (former M.I.S. Organisatie-ingenieurs) who invited him to join MiQ as a consultant in October 1996. In April 1997 he started, part-time, as a Ph. D. student at the faculty of Technology Management at the Eindhoven University of Technology. The thesis in front of you is the result of this Ph.D. research project.

Since January 2001 he joined MiQ as a full-time consultant, which has become part of EDS eSolutions since March 2002. At EDS eSolutions he continues to work in the field of (Rapid)PDM implementations and best practices in New Product Development.

Index of subjects

R

V

Product Data Management as enabler for Concurrent Engineering

Controlling the flow of preliminary information in product development

Changes in competition, customer demands, and technology are the forces that have created a competitive imperative for speed, efficiency and high quality in product development. This research focuses on one particular aspect of improving product development processes: cycle time. There are several ways to speed up the product development process, one of them is Concurrent Engineering and is studied in this research.

The main objective of Concurrent Engineering is to shorten the development time of the product by overlapping phases in the product development process. Starting on preliminary information, downstream processes already start on preliminary information that is released by upstream processes. Starting on preliminary information, however, makes it more complex to manage the product development process. Moreover, it increases the risk which can cause iterations in the development process resulting in extra cost and delays Therefore it is important to release preliminary information in a controlled manner and to communicate changes as quickly as possible to avoid iterations.

Information technology can act as an enabler to achieve this. Therefore in this research it is studied to what extent Product Data Management functionality acts as an enabler for starting on preliminary information. This resulted in a theory that is validated by two case studies, in the process and aerospace industry, which are described in this thesis. Companies can use this theory to design their own overlapped product development process and specify the PDM functionality that they need to control it. The goal of such a redesigned process is a reduction in cycle time to create a competitive advantage.

Beta

Research School for Operations Management and Logistics

Technische Universiteit Eindhoven

Department of Technology Management