4 PRACTICAL EXAMPLES

"Without the element of uncertainty, the bringing off of even the greatest business triumph would be a dull, routine, and eminently unsatisfying affair."
-- J. Paul Getty --

4.1 Introduction

The exploration of the study phenomenon in Chapter 3 led to the identification of a series of questions as input for the exploratory case studies. The findings of these studies, providing answers to questions raised, are presented in this Chapter.

In Section 4.2, the case study approach is discussed. In Section 4.3, a description and detailed overview of the findings are presented for each individual case. Hereafter, in Section 4.4 all case study questions raised in Chapter 3 are answered, using the combined findings of the individual cases. In Section 4.5, the aggregated results are presented. Additional requirements for the adoption of such a methodology are discussed in Section 4.6, and the Chapter ends with a summary and conclusions in Section 4.7.

4.2 Case Study Approach

Case study environments were selected based on the following criteria:

- The selected organization has previously released a significant number of substantial software products. This criterion is used to avoid the conduct of case studies in learning environments, where it is less likely to obtain consensus between informants and supporting documentary evidence.
- Each selected case [consisting of one or more projects] has recently released a software product, or is executing the final tests before product release. This criterion is used to reveal the initial objectives of the project, how they were addressed during product development, and how the software release decision was, or would be, taken.
- Presence of trade-offs between time-to-market, budget, functional requirements and non-functional requirements, including reliability and maintainability.
- Varied case study environments: different software manufacturer types [custom system written in-house versus commercial software] from different countries[49] [X and Y], projects developing different product versions [new product versus new version of existing product], operating at different process maturity [CMM] levels. See Figure 4-1.

Case Id.	Manufacturer type	Product version	Country	CMM level
A	Custom system written in-house	New product	X	1
B	Commercial software	New product	Y	1-2
C	Commercial software	Existing product	Y	3
D	Custom system written in-house	Existing product	Y	2
E	Commercial software	New product	Y	3
F	Custom system written in-house	Existing product	X	2-3
G	Custom system written in-house	New product	X	2-3

Figure 4-1: Overview Case Studies

[49] On request of the participating organizations the names of the countries are not revealed.

A case study protocol was written and reviewed prior to conducting the first case study. This case study protocol is in Appendix A. In Appendix B an example of a case study questionnaire is given. The actual questionnaire used in each case study was adapted to the specific characteristics of each organization. In Appendix C a cross-reference between the identified case study questions in Chapter 3 and the questionnaire in Appendix B can be found.

In the following Section, the results of the exploratory case studies are described, with answers to questions raised in Chapter 3. The names of the participating organizations are not revealed, as the selected projects and the organizational processes are, in most cases, considered of strategic importance. For the validity and reliability of results, these anonymous descriptions are not considered a problem. The findings in each case study originate from the case study report, approved by the participating organization. In this thesis the interpretation of the data by the researcher starts in Section 4.4 with a discussion on, and comparison of, the statements in the case study reports. The thesis therefore remains auditable (Berghout 1997, p.52; based on Boskma and Herweyer 1988).

4.3 Case Study Results

4.3.1 Case Study A

The participating organization is a leading European insurance company. It offers private persons and small/medium-sized corporations a wide range of non-life insurance solutions through traditional, and alternative, distribution channels. Its principal market units are in Europe, with a particular focus on its home country where it is the market leader. Outside Europe, the insurance company operates through several established regional insurance companies. In its home market, the insurance company is the leading all-line carrier, with the largest market share and a close-knit service network.

Selecting people from the different departments compose project teams, with the overall project leader a representative from the Management department. Several sub-projects are defined; responsible for the successful realization of a part of the overall product. Some central teams work for the different sub-projects and may have an additional assignment at project level. The 'IT Management'-team, for example, coordinates architectural issues, integration and test management, and implementation/training.

The selected case is the development of the first version of a new administrative system to handle non-life related insurance claims. The reason for investing is to increase internal organizational efficiency by offering a stable platform, enabling future enhancements. The associated investment level was high [> Euro 10M] as is the perceived strategic value of the release decision [prospect of large financial loss outcomes]. Many new technologies are introduced [hardware infrastructure, development environment] as well as development processes and a new development methodology [RUP]. The organization's process capability is estimated to be CMM level 1 [no institutionalized processes at project level, see Section 1.3.2]. Customers of the product are local agents within the same organization.

For the different project phases, the following findings were noted, with the detailed findings summarized in Figure 4-2:

1. *Product Definition*. Although a product development strategy was defined, it was incomplete, not documented and no consensus regarding priorities was found among informants [non-functional requirements not addressed, although reliability and maintainability were considered important]. The product development strategy was not

derived from a business strategy [not found] or a business case [not available] and different alternatives were not explicitly considered. The initial estimates for schedule and budget were expertise-based, without the availability of historical data from the organization concerned, or the use of external industry averages. Initial estimates were too optimistic, with substantial schedule and budget overruns.

2. *Product Design, Product Realization and Testing*. Reliability and maintainability were not specified, and not addressed, during product design and product realization, although the product was based on a recommended architecture. Reliability was evaluated as an important release criterion during testing, but no explicit attention was paid to maintainability.
3. *Product Release*. There was no formal process available to support the release decision-making process, and the procedure followed can best be described as *ad hoc*. The project leader at operational level took the decision, consulting representatives of the end-users. The decision type is characterized as non-routine, with complete uncertainty, discussed and postponed repetitively, and non-compensatory with reliability as the stop criterion used, while different release alternatives were not explicitly considered. The release date postponements were mainly due to lack of progress in freezing and implementing the functional requirements, and problems in reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was incomplete and of poor quality. The decision to release the product was taken when major defects found during system testing were resolved, although it was known that additional testing would probably reveal more major defects.
4. *Product Rollout*. After the product was released, defects were not archived in a central repository to enable future analysis, with the objective of repairing process deficiencies, and it was not planned to evaluate the business case, or project, afterwards.

Aspect	Description
General	
Organization	- Participating organization is a leading European insurance company.
Manufacturer type	- Custom systems written in-house.
Business strategy	- There was no overall business strategy found.
Process capability	- Estimated CMM level: 1 [no institutionalized processes at project level].
Selected case(s)	- The selected project had developed the first version of a new administrative system to handle non-life related insurance claims. The reason to invest was to increase internal organizational efficiency.
Product characteristics	- Version: first; investment level: high [> Euro 10M]. - Release decision: high strategic value. - Introduction of new hardware infrastructure, development environment, processes, and methodology (RUP). - Time horizon: long-term.
Market characteristics	- Principal market units are in a Western European country. - Customers are local agents within the same organization.
1. Product Definition	
Stakeholders	- Officially: Steering committee at tactical level. - In practice: project leader and representatives from end-users.
Product development strategy	- Defined: time-to-market, development cost, functional requirements, and compliance to external standards [prescribed architecture]. - Non-functional requirements like reliability and maintainability, not defined, although considered important.
Priorities	- Not documented. - No consensus among informants. - For future versions: lead-time expected to become less important.
Degree of changes	- Dynamic [release date postponed several times: schedule/budget overruns].
Business case	- Not available.
Estimation method	- Expertise-based [mathematical, parameterised models not known].
Trade-offs	- No different product development strategies considered.

2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture.
Maintainability	- Not deployed to low-level components in the defined software architecture.
Trade-offs	- No different product design alternatives considered.
Methods	- Unknown, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Officially: Steering committee at tactical level. - Reality: project leader/end-user representatives [higher management informed].
Formal process	- Documented: no. - Followed: no.
Type	- Non-routine, non-recurring [high stress level due to time pressure]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Not complete. - Poor quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable and very uncertain.
Environment	- Open [not all variables known].
Trade-offs	- No different release alternatives explicitly evaluated, other than postponing the release date due to lack of progress in freezing and implementing the functional requirements, and reaching a ‘sufficient’ level of reliability.
Models of choice	- None used.
Outcome	- Very uncertain: operational cost not known.
4. Product Rollout	
Measurements	- Defect repository in place: no. - Analysis of defects: no. - Removal of process deficiencies based on analysis results: no.
Evaluation	- No evaluation of business case or project planned.

Figure 4-2: Main Findings ~ Case Study A

4.3.2 Case Study B

The participating organization is one of the world’s leading companies in the area of document management. In advanced research centres and high-tech production facilities the company develops products and services for the efficient and effective exchange of information, comprising products for the reproduction, presentation, distribution and management of documents. The range of products and services offered is characterized by a recognized high quality, based on reliability, productivity, durability, ease of use and environmental friendliness.

The organization distinguishes different business units, each headed by a Director, who reports to the Board. Within each business unit several business group can be found, each of them addressing a specific market with a specific line of business. The Portfolio Management Committee (PMC) at business unit level defines the product roadmap or program, and approves projects by approving Project Definitions. A Project Definition defines the project’s business case: goals in terms of product needs [functional and non-functional], expected cost price, budget and timing. The Program Steering Committee [PSC] is responsible for implementing the product roadmap or program [i.e. the total project portfolio]. A separate Project Committee [PC] is defined to control each project. The organization uses a matrix organisation for R&D activities. The R&D organisation distinguishes Research, Development and Engineering as separate development phases. Projects are multi-disciplinary: electronic, mechanical, chemical, physical, software, industrial design and development/engineering disciplines are involved. They have no budget, but a temporary assignment with defined goals. As long as a project still exists, it normally takes care of both corrective maintenance and further enhancements. The functional departments assign budget and resources to the projects. Maintenance of released products is also the responsibility of departments once a project ends.

The selected case was the development of the first version of a new printing system. The main reason for investing was to enter the market with a new product. The associated investment level was high [> Euro 10M] as is the perceived strategic value of the release decision, with prospective large financial loss outcomes. Many new product technologies were introduced. The organization's process capability was estimated to be CMM level 1-2 [limited institutionalized processes at project level]. A high level of competition, with few competitors, characterizes the market in which the organization is operating, with customers mainly professional businesses.

The following findings were noted, with detailed findings summarized in Figure 4-3:

1. *Product Definition*. Although a product development strategy was defined and documented, it was not comprehensive [development costs not addressed, maintainability not addressed, although considered important]. No consensus on priorities found among informants. The product development strategy was derived from a business strategy and business case, but alternatives were not considered. The initial estimates for schedule and budget were expertise-based, using personal experiences from the estimators. The initial estimates turned out to be optimistic, leading to substantial schedule and budget overruns.
2. *Product Design* and *Product Realization and Testing*. Reliability and maintainability were not explicitly addressed during product design and product realization, although the product was based on a recommended architecture. Reliability was evaluated as an important release criterion during testing, but no explicit attention was given to maintainability.
3. *Product Release*. There was a formal process available to support the release decision-making process, which was followed, and a steering committee at tactical level took the decision. The decision type was characterized as non-routine, with complete uncertainty, discussed and postponed repetitively, and non-compensatory, with reliability as the most important stop criterion, but release alternatives were not considered, nor explicitly evaluated. The release date was postponed several times due to lack of progress in freezing and implementing the functional requirements, technical problems during implementation, and problems with reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was incomplete and of average quality. The decision to release the product was taken when the major defects found during system testing were resolved.
4. *Product Rollout*. After the product was released, defects were archived in a central repository, but an analysis of the reported defects with the objective of removing process deficiencies was not foreseen. It was not planned to evaluate the business case or project afterwards.

Aspect	Description
General	
Organization	- Organization is one of the world's leading companies in document management.
Manufacturer type	- Commercial software.
Business strategy	- In order of priority: 1. Time-to-market, 2. Functional requirements, 3. Cost price.
Process capability	- Estimated CMM level: 1-2 [limited institutionalized processes at project level].
Selected case[s]	- The selected project had developed the first version of a new printing system. - Reason to invest was to enter the market with a new product.
Product characteristics	- Version: first version of a new product [not only software]. - Investment level: high [> Euro 10M]. - Release decision: high strategic value. - Introduction of new product technologies. - Time horizon: long-term.
Market characteristics	- High level of competition from few competitors. - Products are sold worldwide. - Customers are mainly professional users [business-to-business].
1. Product Definition	
Stakeholders	- Steering committee at tactical level.
Product development strategy	- Defined: time-to-market, functional requirements, non-functional product requirements, cost price, compliance to external standards. - Not defined: development cost [considered to be of less importance]. - Not defined: maintainability as a non-functional product need, although considered important.
Priorities	- Documented. - No consensus among informants. - For future versions: time-to-market expected to be less important.
Degree of changes	- Dynamic [release dates postponed: considerable schedule/budget overruns].
Business case	- Initially available [no financial or other method used], not updated.
Estimation method	- Expertise-based [mathematical, parameterised models not known].
Trade-offs	- No different product development strategies considered.
2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture.
Maintainability	- Not deployed to low-level components in the defined software architecture.
Trade-offs	- No different product design alternatives considered.
Methods	- Unknown, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Steering committee at tactical level [higher management informed].
Formal process	- Documented: yes. Level of completion: limited. - Followed: yes.
Type	- Non-routine, non-recurring [medium to high stress level due to time pressure]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, cost price, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Not complete. - Average quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable and very uncertain.
Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in freezing and implementing the functional requirements, technical problems during implementation, and

	reaching a 'sufficient' level of reliability.
Models of choice	- None used.
Outcome	- Uncertain: operational cost not known.
4. Product Rollout	
Measurements	- Defect repository in place: limited. - Analysis of defects: no. - Removal of process deficiencies based on analysis results: no.
Evaluation	- No evaluation of business case or project planned.

Figure 4-3: Main Findings ~ Case Study B

4.3.3 Case Study C

The participating organization supplies operators and service providers around the world, with end-to-end solutions, for all existing systems in mobile and broadband Internet. The solutions include network infrastructure, access equipment and terminals, application enablers and global services.

The organization sells products and services to the local markets [Benelux] via the Market Unit. Research and development activities take place in the so-called Intercompany Unit. They are not restricted to the local markets and can be inter-company wide. A strategic product manager can order a design unit to execute the assignment to develop a product. The unit Project Office is responsible for taking care of all assignments. The actual development of the software is done in the design unit, where different departments exist for specific product portfolios. The project office consists of provisioning managers, who manage the projects related to the strategic roadmap. The design unit supplies a project management team, consisting of, at least, a project manager, quality coordinator and technical coordinator.

The selected case was the development of a new version [9th revision] of a software platform for a digital telephone exchange, offering the facility to load intelligent network services. The reason to invest was to offer enhanced functionality. The associated investment level was medium [> Euro 1M and < Euro 10M] and the perceived strategic value of the release decision, including the presence of prospective large financial loss outcomes. The official organization's process capability was CMM level 3. A high level of competition from few competitors characterizes the market in which the organization is operating with customers mainly network operators.

For the different project phases, the following findings were made, as also summarized in Figure 4-4:

1. *Product Definition*. A product development strategy was defined, which was complete and documented. The product development strategy was derived from a business case, but no overall business strategy was found, and alternatives were not explicitly considered. The initial estimates for schedule and budget were expertise-based, using past experience in the organization, but not using external industry averages. The initial estimates turned out to be optimistic, leading to a schedule and budget overrun.
2. *Product Design* and *Product Realization and Testing*. Reliability and maintainability were not explicitly addressed during product design and product realization, although the product was based on a recommended architecture. Reliability was evaluated as an important release criterion during inspections and testing, whereas maintainability was not addressed. However, it was stated that stringent coding rules are regarded as an indirect means to assure high maintainability.
3. *Product Release*. There is a formal process available to support the release decision-making process, which was also followed. A steering committee at tactical level took the decision. The decision type was characterized as routine, with limited uncertainty,

discussed and postponed repetitively, and non-compensatory with reliability as the most important stop criterion, but release alternatives were not considered. The release date was postponed several times due to lack of progress in freezing and implementing the functional requirements, and problems with reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was complete and of a high quality. The decision to release the product was taken when the major defects found during system testing were resolved.

4. *Product Rollout*. After the product was released, defects were archived in a central repository, with limited analysis aiming to remove process deficiencies. It was not planned to evaluate the business case or project later.

Aspect	Description
General	
Organization	- Organization is one of the world's leading companies in telecommunications.
Manufacturer type	- Commercial software.
Business strategy	- No overall business strategy found.
Process capability	- Officially assessed at CMM level: 3.
Selected case[s]	- The product developed was a software platform on a digital telephone exchange, offering the facility to load intelligent network services. The reason to invest was to offer enhanced functionality.
Product characteristics	- Version: 9th revision of an existing product [near end of life]. - Investment level: medium [> Euro 1M, < Euro 10M]. - Release decision: high strategic value. - Time horizon: long-term.
Market characteristics	- High level of competition from few competitors. - Products are sold worldwide. - Customers are mainly network operators.
1. Product Definition	
Stakeholders	- Steering committee at tactical level.
Product development strategy	- Defined: lead-time, budget, functional requirements, non-functional requirements, and compliance to internal and external standards.
Priorities	- No priorities: all criteria had to be met without exceptions. - Reliability and maintainability were extremely important.
Degree of changes	- Attainable [release date postponed several times].
Business case	- Initially available [no financial or other method used], not updated.
Estimation method	- Expertise-based [mathematical/parameterised models known, not used].
Trade-offs	- No different product development strategies considered.
2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture, but very detailed design and coding rules available
Maintainability	- Not deployed to low-level components in the defined software architecture, but very detailed design and coding rules available.
Trade-offs	- Different product design alternatives considered.
Methods	- Unknown, not used.
2b. Product Realization and Testing	
Reliability	- Evaluated via inspections, and integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Steering committee at tactical level.
Formal process	- Documented: yes. Level of completion: medium. - Followed: yes.
Type	- Routine, recurring [low stress level]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, test results as an indication for reliability, compliance to internal and external

	standards. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Complete. - High quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable and very uncertain.
Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in freezing and implementing the defined functional requirements, and reaching a ‘sufficient’ level of reliability.
Models of choice	- None used.
Outcome	- Relatively certain: minor changes implemented with respect to previous version and development processes have been prescribed in detail.
4. Product Rollout	
Measurements	- Defect repository in place: limited. - Analysis of defects: limited. - Removal of process deficiencies based on analysis results: limited.
Evaluation	- No evaluation of business case or project planned.

Figure 4-4: Main Findings ~ Case Study C

4.3.4 Case Study D

Participating organization is an independent business unit within a multi-national, offering flexible staffing, projects and consultancy. The project organization develops product software, both for products and for support during production of these products. Products are more and more interconnected via networks. The share of connectivity-related software is therefore increasing. Connections to the Internet are often required. The organization is heavily involved in creating products for digital audio and video areas, e.g. CD, DVD-RW, digital-TV and Set Top Boxes. Web-TV and Internet-Radio are logical extensions in this field, including corresponding e-business applications. The software created varies from embedded software on dedicated hardware to complex technical applications on multiple-processor based systems. Besides real-time operating systems, Unix or MS-Windows, including look-alikes and derivates, are often applied. The share of Linux and Embedded Windows-NT is increasing. The software architecture used is often based on multiple abstraction layers, each describing the corresponding functionality. One recognizes hardware, operating system and device driver layers. The corresponding database and application software is also independently created. Corresponding tools and systems often support user interface development. More and more product-related software can be updated remotely and the product can be controlled remotely. Corresponding interfaces are mainly based on TCP/IP and web-technology. Examples in this area are remote service and control of products, including corresponding e-business applications.

The selected case was the development of a major revision of an existing software product that provides facilities to transfer large volumes of data via a dedicated network. The associated investment level is low [< Euro 0.5M] as well as the perceived strategic value of the release decision strategic value, no prospective large financial loss outcomes. The organization’s process capability is CMM level 2. The product was developed for an external company [principal] to be further maintained by that company after the product release. The reason for the principal investing is the replacement of an existing system, so as to enable future enhancements.

For the different project phases, the following findings were noted, with detailed findings in Figure 4-5:

1. *Product Definition*. A product development strategy was defined, which was complete and documented, however no consensus was found among the informants. The product development strategy was derived from a business case, but overall business strategy was not found, and alternatives were not explicitly considered. The initial estimates for schedule and budget were expertise-based, using experience within the organization but not using external industry averages. The initial estimates were optimistic, leading to a schedule and budget overrun.
2. *Product Design* and *Product Realization and Testing*. Reliability and maintainability were not explicitly addressed during product design and product realization. Reliability was evaluated as an important release criterion during testing, while maintainability was not addressed.
3. *Product Release*. There was a formal process available to support the release decision-making process, which was partially followed, although its level of completion was limited. A steering committee at tactical level took the decision. The decision type was characterized as routine, despite a high level of uncertainty, discussed and postponed repetitively, and non-compensatory with reliability as the most important stop criterion, but release alternatives were not considered, nor explicitly evaluated. The release date was postponed several times due to lack of progress in freezing and implementing the functional requirements, and problems in reaching a ‘sufficiently’ reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was complete and of high quality. The decision to release the product was taken when the major defects found during system testing were resolved.
4. *Product Rollout*. After the product was released, defects were archived in a central repository, with limited analysis aiming to remove process deficiencies. It was not planned to further evaluate the business case or project afterwards.

Aspect	Description
General	
Organization	- Participating organization is an independent business unit within a multi-national, offering flexible staffing, projects and consultancy.
Manufacturer type	- Custom systems written on contract.
Business strategy	- There was no overall business strategy found.
Process capability	- Officially assessed at CMM level: 2.
Selected case[s]	- The product developed was a software product that provides facilities to transfer large volumes of data via a dedicated network. Product is developed for an external company [principal] and will be further maintained by that company after product release. The reason for the principal to invest was the replacement of an existing system to enable future enhancements.
Product characteristics	- 4th version of an existing product [redesign]. - Investment level: low [< 0.5M Euro]. - Release decision: low strategic value. - Time horizon: short-term to mid-term.
Market characteristics	- Customers are businesses.
1. Product Definition	
Stakeholders	- Steering committee at tactical level.
Product development strategy	- Defined: time-to-market, budget, functional requirements, non-functional requirements, [including reliability and maintainability], compliance to internal standards.
Priorities	- Documented in project management plan. - No consensus among informants. - For earlier versions: time-to-market expected to have been more important.
Degree of changes	- Attainable [release date postponed several times].
Business case	- Initially available [no financial or other method used], not updated.
Estimation method	- Expertise-based [mathematical, parameterised models known, not used].
Trade-offs	- No different product development strategies considered.

2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture.
Maintainability	- Not deployed to low-level components in the defined software architecture.
Trade-offs	- Different product design alternatives considered.
Methods	- Known, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release [not yet taken place, and the expectations of the informants are summarized here]	
Stakeholders	- Steering committee at tactical level.
Formal process	- Documented: yes. Level of completion: limited. - Followed: partially.
Type	- Routine, recurring [medium stress level]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Near complete. - Average quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable and very uncertain.
Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in freezing and implementing the defined functional requirements, and reaching a 'sufficient' level of reliability.
Models of choice	- None used.
Outcome	- Uncertain: operational cost not known.
4. Product Rollout	
Measurements	- Defect repository in place: yes. - Analysis of defects: limited. - Removal of process deficiencies based on analysis results: limited.
Evaluation	- No evaluation of business case or project planned.

Figure 4-5: Main Findings ~ Case Study D

4.3.5 Case Study E

The participating organization is the world's leading supplier of instrumentation and software for X-ray diffractometry and X-ray fluorescence spectrometry, with more than a half century of experience. The company offers X-ray analytical equipment for industrial and scientific applications, as well as for the semiconductor market.

The organization distinguishes two Lines Of Business [LOB's]: 'Industrial and Scientific Research' and 'Process and Quality Control'. Each LOB defines a road map, which is reviewed at a strategic level. A business case is defined for each project: goals in terms of product needs [functional and non-functional], expected cost price, budget and lead-time. Projects are multi-disciplinary: electronic, mechanical, physical disciplines [Hardware Development group] and software disciplines [Software Development group] are involved. Each project has a temporary assignment with defined goals. The hardware and software development groups assign budget and resources to the projects. Maintenance of released products is also the responsibility of departments [in exceptional cases, dedicated projects are defined].

The selected case was the development of a new product for the calculation and analysis of measured values obtained by X-ray fluorescence. Reason to invest: entering the market with a new product. The associated investment level was high [> Euro 10M] as was the perceived strategic value of the release decision, in the presence of large prospective financial loss outcomes. Many new product technologies were introduced. The official organization's process capability is CMM level 3. A high level of competition from few competitors characterizes the market in which the organization is operating, with customers mainly professional businesses.

For the different project phases the following findings were made, with details summarized in Figure 4-6:

1. *Product Definition*. A product development strategy was defined, which was complete and documented, but no consensus regarding the priorities was found among informants. The product development strategy was derived from a business strategy [road map] and business case, but alternatives were not explicitly considered. The initial estimates for schedule and budget were expertise-based, using past experience of the organization but not using external industry averages. The initial estimates turned out to be too optimistic, leading to substantial schedule and budget overruns.
2. *Product Design* and *Product Realization and Testing*. Reliability and maintainability were not addressed during product design and product realization [although the product was based on a recommended architecture]. Reliability was evaluated as an important release criterion during testing, whereas maintainability was not addressed.
3. *Product Release*. There was a formal process available to support the release decision-making process, which was partially followed. A steering committee at tactical level took the decision. The decision type was characterized as non-routine, with complete uncertainty, discussed and postponed repeatedly, and non-compensatory with reliability the most important stop criterion. Different release alternatives were not explicitly evaluated. The release date was postponed several times, due to lack of progress in freezing and implementing the functional requirements, the encounter of technical problems during implementation and problems in reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was incomplete and of average quality. The decision to release the product was taken when major defects found during system testing were resolved.
4. *Product Rollout*. After the product was released, defects were archived in a central repository, with limited analysis aiming to remove process deficiencies. It was not planned to further evaluate the business case or project afterwards.

Aspect	Description
General	
Organization	- Participating organization is one of the world's leading suppliers of instrumentation and software for X-ray analysis.
Manufacturer type	- Commercial software.
Business strategy	- In order of priority: 1. Quality, 2. Lead-time, 3. Functionality, 4. Budget.
Process capability	- Officially assessed at CMM level: 3.
Selected case[s]	- The product calculates and analyses the measured values obtained by X-ray fluorescence. Reason to invest was to enter the market with a new product.
Product characteristics	- First version of a new product [not only software]. - Investment level: medium [> Euro 1M, < Euro 10M]. - Release decision: high strategic value. - Introduction of new product technologies. - Time horizon: long-term.
Market characteristics	- High level of competition from few competitors. - Products are sold worldwide. - Customers are professional users [business-to-business].

1. Product Definition	
Stakeholders	- Steering committee at tactical level.
Product development strategy	- Defined: time-to-market, budget, functional requirements, non-functional product requirements [including reliability and maintainability], cost price, compliance with external and internal standards.
Priorities	- Documented (see product development strategy). - No consensus among informants. - Time-to-market expected to be less important for future versions.
Degree of changes	- Dynamic [release date postponed several times: considerable schedule/budget overruns].
Business case	- Initially available [no financial or other method used], not updated.
Estimation method	- Expertise-based [mathematical, parameterised models not known].
Trade-offs	- No different product development strategies considered.
2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture.
Maintainability	- Not deployed to low-level components in the defined software architecture.
Trade-offs	- No different product design alternatives considered.
Methods	- Known, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Steering committee at tactical level [higher management informed].
Formal process	- Documented: yes. Level of completion: limited. - Followed: partially.
Type	- Non-routine, non-recurring [medium to high stress level due to time pressure]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Not complete. - Average quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable and uncertain.
Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in freezing and implementing the defined functional requirements, technical problems during implementation, and reaching a ‘sufficient’ level of reliability.
Models of choice	- None used.
Outcome	- Very uncertain: operational cost not known.
4. Product Rollout	
Measurements	- Defect repository in place: limited. - Analysis of defects: limited. - Removal of process deficiencies based on analysis results: limited.
Evaluation	- No evaluation of business case or project planned.

Figure 4-6: Main Findings ~ Case Study E

4.3.6 Case Study F

The participating organization develops and maintains an electronic trading platform, enabling the fully automated execution, clearing and settlement of all exchange transactions. The platform offers an order-driven market, off order book trading, reporting functions as well as automatic clearing and settlement. This real-time processing of stock market transactions is achieved by a group of subsystems interacting with each other in a network environment. It is the home exchange and key market for major international companies. Over 100 participants from the home country and other countries, including some of the most important global players in the financial markets, trade on this exchange market.

From the strategic planning process with a horizon of several years, the management boards draws up a business plan and related financial budget for the next year in the following subsequent steps:

- ❖ Ideas for new projects are investigated and assessed as to how they fit in the overall strategy.
- ❖ Existing projects are roughly calculated for the coming year in terms of effort, costs and investments. From the selected new projects and existing projects the needed investments are derived, as well as the staff profile, leading to the project portfolio.
- ❖ The project investments are combined with additional costs and are to be formally approved by the board of directors.

The responsible department starts the selected projects on planned dates. Projects that are not part of the business plan may eventually be started after formal approval by the management board. The business plan or project portfolio is reviewed four times a year. Attention is given to the actual status of running projects and the planned data of projects still to start. Progress of all projects running is measured on a weekly basis. The management board reviews all projects every four weeks and ensures that current and planned activities, as part of the project portfolio, remain aligned with the overall business plan.

The case selected was the development of a software platform, offering an electronic trading facility for stock market transactions. The reason to invest was to offer enhanced functionality. The associated investment level was medium [> Euro 1M], whereas the perceived strategic value of the release decision was high, in the presence of large prospective financial loss outcomes. The organization's process capability was estimated to be CMM level 2-3 [institutionalized processes at project level and limited at organizational level]. Customers are banks and investment companies, as participants in the exchange.

The following findings were noted, with detailed findings summarized in Figure 4-7:

1. *Product Definition*. A product development strategy was defined, which was both comprehensive and documented. The product development strategy was further derived from a business strategy [annual business plan] and business case, but different alternatives were not explicitly considered. The initial estimates for schedule and budget were expertise-based, using past experience of the organization but not using external industry averages. The initial estimates turned out to be slightly optimistic, leading to a schedule overrun.
2. *Product Design* and *Product Realization and Testing*. Reliability and maintainability were not explicitly addressed during product design and product realization, although the product was based on a recommended architecture. Reliability was evaluated as an important release criterion during testing, whereas maintainability was not addressed.
3. *Product Release*. There was a formal process available to support the release decision-making process, which was partially followed, although its level of completion was limited. A steering committee at tactical level took the decision. The decision type was characterized as routine, with limited uncertainty, discussed and postponed repetitively, and non-compensatory with reliability as the most important stop criterion. Different release alternatives were not explicitly evaluated. The release date was postponed several times due to lack of progress in freezing and implementing the functional requirements, the encounter of technical problems during implementation, and problems with reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development

documentation was near complete and of average to high quality. The decision to release the product was taken when the major defects found during system testing were resolved.

4. *Product Rollout*. After the product was released, defects were archived in a central repository, without analysis aiming to remove process deficiencies. It was not planned to further evaluate the business case or project afterwards.

Aspect	Description
General	
Organization	- Participating exchange organization brings together participants, issuers and investors in an efficient and transparent securities market.
Manufacturer type	- Custom systems written in-house.
Business strategy	- There was no overall business strategy found.
Process capability	- Estimated CMM level: 2-3 [institutionalized processes at project level and limited at organizational level].
Selected case[s]	- The product developed is a software platform, offering an electronic trading facility for stock market transactions. The reason to invest was to offer enhanced functionality.
Product characteristics	- Version: 6th revision of existing product; investment level: medium [> Euro 1M]. - Release decision: high strategic value. - Time horizon: long-term.
Market characteristics	- Limited competition; customers are banks and investment companies, being participants of the exchange.
1. Product Definition	
Stakeholders	- Steering committee at tactical level.
Product development strategy	- Defined: time-to-market, budget, functional product needs, non-functional products needs [including reliability and maintainability].
Priorities	- Functional and non-functional requirements. - Not explicitly documented.
Degree of changes	- Attainable [release date postponed several times].
Business case	- Initially available [no financial or other method used], not updated. - The business case only mentioned the development cost due to the inability to allot benefits to a specific product release.
Estimation method	- Expertise-based [mathematical, parameterised models not known].
Trade-offs	- No different product development strategies considered.
2a. Product Design	
Reliability	- Not deployed to low-level components in the defined software architecture.
Maintainability	- Not deployed to low-level components in the defined software architecture.
Trade-offs	- No different product design alternatives considered.
Methods	- Unknown, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Steering committee at tactical level [higher management informed].
Formal process	- Documented: yes. Level of completion: limited. - Followed: yes.
Type	- Non-routine, non-recurring [medium stress level]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, compliance with internal and external standards, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Near complete. - Average to high quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable.

Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in reaching a 'sufficient' level of reliability.
Models of choice	- None used.
Outcome	- Relatively certain: minor changes implemented on previous version.
4. Product Rollout	
Measurements	- Defect repository in place: yes. - Analysis of defects: no. - Removal of process deficiencies based on analysis results: no.
Evaluation	- No evaluation of business case or project planned.

Figure 4-7: Main Findings ~ Case Study F

4.3.7 Case Study G

The participating organization is a leading global financial services company. The organization aims to continuously invest in its core businesses and technology; to deliver a strong performance for its clients; and demonstrate responsibility towards the environment and the communities in which it operates. The organization's strategy is to build its leadership as one of the top five global financial services companies by focusing on two core businesses in key markets: asset gathering/management and financial intermediation.

Two cases were studied in this organization. The first selected case was the development of a set of various mortgage-related products. This project has a long history that goes back to 1982. In the nineties, several attempts were to start a project to renew the existing application. For various reasons, these attempts failed. In March 2001, the project under review was (re-) started, which delivered the first release of a new product in November 2002, with customers for the product local agents within the same organization The associated investment level was high [> Euro 10M] as well as the perceived strategic value of the release decision, in the presence of large prospective financial loss outcomes. The project was financed both by the business division and the IT division. Although its calculated Net Present Value is negative, the upgrade to a new product was felt to be compulsory due to changed user requirements and technological developments. The second case was a project to offer a general platform for other e-commerce products. In this sense, this infrastructure project did not deliver a product for end-users. The primary focus in this case study has been on the first project, whereas the second project was used for verification. Through this approach, it is assumed results can be generalized. The organization's process capability was estimated to be CMM level 2-3 [institutionalized processes at project level and limited at organizational level].

The following findings were noted, with details summarized in Figure 4-8:

1. *Product Definition*. The product development strategy was both defined and comprehensive, but was not documented and no consensus on the priorities was found among informants. The product development strategy was derived from a business case, but overall business strategy was not found, and different alternatives were considered. The initial estimates for schedule and budget were expertise-based, using past experience in the organization but not using external industry averages. The initial estimates turned out to be too optimistic, leading to substantial schedule and budget overruns. The main reason for these overruns was difficulty in 'freezing' requirements.
2. *Product Design* and *Product Realization and Testing*. Both reliability and maintainability were addressed during product design and product realization. Reliability was evaluated as an important release criterion during testing, whereas maintainability was not addressed.
3. *Product Release*. There was a formal process available to support the release decision-making process, which was also followed, although its level of completion was limited.

A steering committee at tactical level took the decision. The decision type was characterized as non-routine, with complete uncertainty, discussed and postponed repetitively, and non-compensatory with reliability as the most important stop criterion. Different release alternatives were not explicitly evaluated. The release date was postponed several times due to lack of progress in freezing and implementing the functional requirements, and problems with reaching a 'sufficiently' reliable product. An estimated budget for short-term and long-term maintenance activities was not available. The supporting product and development documentation was near complete and of average to high quality. The decision to release the product was taken when major defects found during system testing were resolved.

4. *Product Rollout*. After the product was released, defects were archived in a central repository, but analysis, with the objective of removing process deficiencies, was not foreseen although it was planned to evaluate the business case afterwards.

Aspect	Description
General	
Organization	- Participating organization is a leading global financial services company.
Manufacturer type	- Custom systems written in-house.
Business strategy	- There was no overall business strategy found.
Process capability	- Estimated CMM level: 2-3 [institutionalized processes at project level and limitedly at organizational level].
Selected case[s]	- The product developed covers a set of various mortgage-related products. Reason to invest was the replacement of an existing system to increase productivity. - In this case study, a second project was also observed. This project developed an IT-infrastructure for other projects, also strategically important and with a high investment level. The findings confirmed the findings made in the reference project.
Product characteristics	- First version of a new software product; investment level: high [> Euro 10M]. - Release decision: high strategic value. - Introduction of new hardware infrastructure. - Time horizon: long-term.
Market characteristics	- No competition. - Customers are local agents within the same organization.
1. Product Definition	
Stakeholders	- Officially: Steering committee at tactical level. - In practice: project leader and end-user representatives.
Product development strategy	- Defined: time-to-market, development cost, functional requirements, non-functional requirements [especially reliability, maintainability and performance] and compliance to internal standards.
Priorities	- Not documented. - No consensus among informants. - For future versions: time-to-market expected to become less important.
Degree of changes	- Dynamic [release date postponed several times: considerable schedule/budget overruns].
Business case	- Initially available [NPV], updated.
Estimation method	- Expertise-based [mathematical, parameterised models not known].
Trade-offs	- Different product development strategies considered.
2a. Product Design	
Reliability	- Deployed to low-level components in the defined software architecture.
Maintainability	- Deployed to low-level components in the defined software architecture.
Trade-offs	- No different product design alternatives considered.
Methods	- Unknown, not used.
2b. Product Realization and Testing	
Reliability	- Only evaluated during integration and system testing.
Maintainability	- Not evaluated.
3. Product Release	
Stakeholders	- Steering committee at tactical level [higher management informed].
Formal process	- Documented: yes. Level of completion: limited. Followed: yes.

Type	- Non-routine, non-recurring [medium to high stress level due to time pressure]. - Complete uncertainty [environment, alternatives]. - Repetitive. - Non-compensatory.
Information	- Available: spent lead-time, spent budget, implemented functional requirements, test results as an indication for reliability. - Not available: maintainability level.
Maintenance budget	- No budget estimate for short-term maintenance effort [corrections]. - No budget estimate for long-term maintenance effort [enhancements].
Documentation	- Near complete. - Average to high quality.
Constraints	- Time and cost.
Quantifiable	- Reliability and maintainability levels not quantifiable.
Environment	- Open [not all variables known].
Trade-offs	- No different explicit release alternatives evaluated, other than postponing the release date due to lack of progress in freezing and implementing the defined functional requirements, and reaching a 'sufficient' level of reliability.
Models of choice	- None used.
Outcome	- Uncertain: operational cost not known.
4. Product Rollout	
Measurements	- Defect repository in place: no. Analysis of defects: no. - Removal of process deficiencies based on analysis results: no.
Evaluation	- Success of business case planned to be evaluated.

Figure 4-8: Main Findings ~ Case Study G

4.4 Answers to the Exploratory Questions

The case study results presented in the previous Section are used to find the answers to the questions raised in Chapter 3. Pattern-matching is used as the analytical strategy (see Section 2.7). A summary of the results is also presented by Sassenburg and Berghout (2004).

4.4.1 Economic Perspective

1. Which product development strategies do different software manufacturer types use?

In Figure 4-9, the product development strategies with the priority order of project objectives found are summarized for the different cases studied. It is indicated if the product development strategy was documented [+] or not [–] and whether the informants showed consensus on their perception of the product development strategy [+] or not [–].

The following observations are made:

- ❖ In four cases [B, C, D, E] the product development strategies were documented, however only in two cases did informants report consensus on documented [C] or perceived [F] priorities. In the cases where no documentary evidence could be found on the product development strategy, one case [F] showed consensus among the informants.
- ❖ Intuitively, a correlation between process maturity level and the availability of a documented and commonly accepted product development strategy was expected. Some supporting evidence was found: the organization with the highest process maturity level [C] matches both conditions [documented and consensus] and the organization with the lowest process maturity level [A] matches none. However, the situation found in, for example, case G [high process maturity level but no matches] is in contrast.
- ❖ For the product release moment, the focus in all cases was on reaching a sufficiently high level of reliability. This is remarkable, as in all cases but two [C, E], reliability was not

explicitly documented or stated within the product development strategy as a high priority issue [in D it was assigned the lowest priority]. However, in all cases it was implicitly the main driver for postponing the release date during the test phases.

- The reasons for delays in the release date were similar in all cases: difficulties in making the functional requirements explicit and complete, changes to the initially frozen requirements, incomplete and/or unrealistic project plans resulting in poor project performance. The results of a study conducted by the Standish Group (1995) reveal similar reasons for project failure, as in Figure 4-10. In three cases [A, B, E], additional complexity arose from the introduction of new technologies.

Case	Product Version	Product Development Strategy		
		Priority Order: high → low	Documented	Consensus
A	First	Unknown	–	–
B	First	Time-to-market, functional requirements, cost price	+	–
C	Subsequent	All	+	+
D	Subsequent	Development cost, functional requirements, performance, portability, time-to-market, maintainability, service maintainability, reliability	+	–
E	First	Reliability, time-to-market, functional requirements, development cost	+	–
F	Subsequent	Functional requirements, non-functional requirements	–	+
G	First	Unknown	–	–

Figure 4-9: Priorities for Product Development Strategy

The first conclusion is that documented and commonly-accepted product development strategies are not common in the cases studied. This is an important conclusion, as having consensus among all stakeholders about the priority setting in a product development strategy is an important factor to enable selection among different alternatives at crucial project milestones, such as project definition, product design and product release. If different stakeholders are not working toward a common goal, then energy that should be focused on meeting that goal could be dissipated.

Reason for Failure		Percentage [%]
1.	Requirements not explicit enough – different interpretations possible	13.1
2.	Users insufficiently involved	12.4
3.	Lack of resources [budget, people]	10.6
4.	Unrealistic expectations	9.9
5.	Insufficient management support	9.3
6.	Requirements unstable [subject to changes]	8.7
7.	Project plans incomplete or lack of sufficient detail	8.1
8.	No more need for the application	7.5

Figure 4-10: Fail Factors in Software Projects
(Standish Group 1995)

The second conclusion is that although not always made explicit, or stated, reliability plays an important role in the cases studied, as it was the most important criterion for postponing the release date during the later project phases of implementation and test.

2. Do law-like generalizations for product development strategies exist, as in Moore's model?

In cases A, B, E and G new products were developed. Time-to-market was considered the most important criterion in case B, but not in case E where reliability was considered most important. For cases A and G, the situation was not clear. In cases C, D and F subsequent versions of existing products were developed. Both functional and non-functional requirements, among which reliability, are considered important.

In practice, all cases revealed an increasing focus on reliability near the end of product development prior to the release decision. This was true both where new products, and subsequent versions of existing products, were developed. In case B for example, it was stated during the interviews, and later discussions about the final case study report, that the organization has a reputation to uphold [that their products are reliable], irrelevant of whether it is a new product or not.

No evidence could be found supporting the validity of Moore's model. It may be valid in some situations, but its validity heavily depends on characteristics of the environment, and the software manufacturer organization. It may well be true that Moore's model applies to mass-markets for commercial software products. Software manufacturers studied here do not operate in these markets.

Other law-like generalizations could not be made, other than two observations. In the first place, all cases studied revealed an increasing focus on reliability near the end of product development, although it was hardly possible to quantify the reliability level obtained. Put differently, whether or not addressed as a high priority in the product development strategy, reliability was always a high priority issue prior to the release decision. The second observation is that, irrespective of whether maintainability was mentioned in the product development strategy or not, its actual value prior to the release decision was not considered.

3. Which market entry strategies do different software manufacturer types use?

In all cases except one [A], business cases were developed as the rationale for the investment decision prior to the start of the related projects. Only in one case [G], did the cost/benefit analysis include the application of a capital budgeting method (NPV). In all cases the initial project definition changed during subsequent project stages, of design, implementation and test, but the initial business case was not updated to reflect the new situation except in case G. Changes in the external environment and unexpected project performance forced all projects to reconsider the initially-planned release date and thus their market entry strategy. The fact that non-functional requirements like reliability and maintainability criteria could not be quantified near the release date also made it difficult, if not impossible, to assess their contribution to costs and benefits in purely financial terms. In none of the cases was the release decision based on a cost-benefit analysis from a strictly economic point of view.

Even if it was possible to assign values to, for example, reliability, it is questioned whether this would enable a cost-benefit analysis. It would require knowledge about the exact relationships between the different project objectives. Suppose the relationships between the four objectives time-to-market, functionality, development cost and reliability were known, it would enable a decision-maker to apply six trade-off rules among these parameters, as in Figure 4-11.

Each circle is related to an objective and represents an equivalent loss value. Having such a scheme available enables a decision-maker to consider different trade-offs. It would, for example, be advantageous if a cut-off of T months from the development time could be met by spending less than C/T money. In reality, such a trade-off model would be far more complex by introducing additional parameters, such as short-term operational cost, [for corrective maintenance], long-term operational cost [for adaptive/perfective maintenance] and revenues.

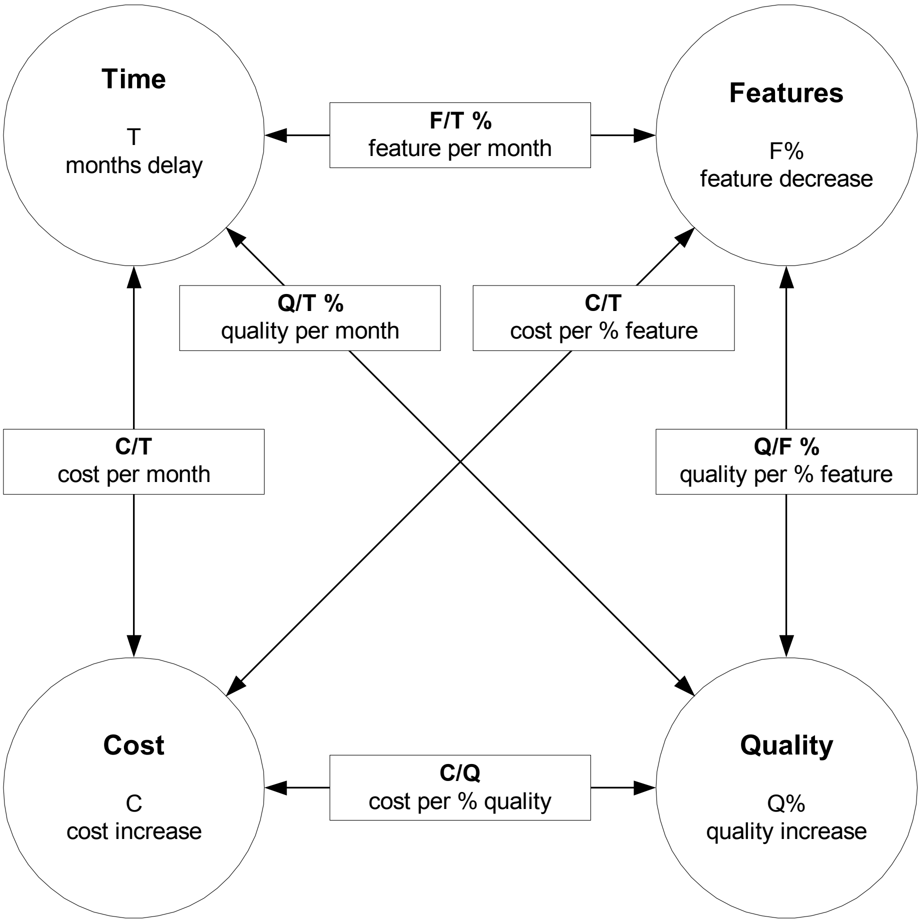

Figure 4-11: Trade-off Rules for Four Parameters
(Smith 1990)

The first conclusion is that, in the study environment, using supporting methods to make a cost-benefit analysis, or investment appraisal, is not common [except in G]. The market entry strategy found in the case studies was to increase the product reliability until it was found to be ‘good enough’. Secondly, it is concluded that a purely quantitative cost-benefit analysis to determine a market entry strategy is difficult. Quantifying all parameters without uncertainties is problematic, as it is unlikely that the exact relationships between all parameters will be known. Given the fact that this applies to all cases studied, it is assumed this will apply to software development in general.

4.4.2 Software Management Perspective

4. Which methods are used to estimate development cost and schedule?

In all cases, expertise-based estimates were made for development cost and schedule. However, in none of the cases was a repository with information from past projects found to support the estimates. None of the organizations mentioned the application of mathematical, parameterised models like COCOMO II or SLIM-Estimate. The existence of such models was in most cases unknown, except in cases C and D.

It is concluded that a gap exists between the results of academic efforts and the application of these results in the software industry; at least in the cases studied. The conclusion might be valid beyond the studied environments, as this gap has been identified before. Poore (2004), for example, states that software engineering suffers a decided lack of direction, whereas other disciplines seem to evolve from theory to practice relatively responsibly. The reason for this gap remains unclear. Does the theory not address practical needs, or does the software industry consciously, or unconsciously, ignore the available results? Related to this, in Section 4.6 some remarks on model adoption will be put forward; considered important for the methodology to be designed.

5. Are different project and design alternatives considered?

In all cases, no different project definition alternatives were explicitly evaluated. Based on available information a project proposal or project plan was developed and reviewed.

Only in case G were different design alternatives considered. In all other cases, either the existing design or architecture was reused, or adapted, [C, D, F], or a new design or architecture was developed without considering different alternatives [A, B, E]. In all cases, supporting methods were neither known nor used: for example, the *Architecture Tradeoff Assessment Method* or *ATAM* (Kazman *et al.* 1998); *Software Architecture Assessment Method* or *SAAM* (DeSimone, and Kazman 1995); *Cost Benefit Analysis Method* or *CBAM* (Asundi *et al.* 2000; Asundi and Kazman 2001); and *Model Based Architecting and Software Engineering* or *MBASE* (Boehm *et al.* 1999).

Again, the important conclusion is the existence of a gap between the results of academic efforts and the application of these results in the software industry, at least in the cases studied, and possibly/probably valid beyond the studied environments. Whether methods for considering different design alternatives would be beneficial was not investigated, however it is assumed that this is likely to be true especially in cases where, for example, a trade-off must be made between conflicting non-functional requirements.

6. To what extent are reliability requirements defined, deployed and evaluated?

In most cases, reliability turned out to be an important project objective, irrespective of whether it was documented, or whether there was consensus among the informants. In most cases, the requirements with respect to reliability were documented, for example, as the Mean Time Between Failures in hours (*MTBF*) or Availability [*MTBF/(MTBF + MTTR)*, where *MTTR* is Mean Time To Repair].

Only in case G was evidence found that, during the design or architecture phase, time and effort was spent to deploy reliability to identified components in the design. In case C detailed design and coding rules were available with the implicit objective of contributing to a high level of reliability.

In all cases, reliability was evaluated by integration and system tests. However, in none of the cases were software defect prediction models, or software defect estimation models, used. Both developers and representatives from other departments analysed the number of reported problems over a certain period of time. When the number stabilized or remained below a certain threshold, the software was considered ‘good enough’. This closely resembles the assumption made in Section 3.3.2 that software manufacturers consciously, or unconsciously, continue testing until all known critical problems are removed, whereafter further testing is considered to be more harmful than helpful.

It is concluded that reliability is normally an important non-functional requirement, but that the deployment of its defined value to design components as well as the determination of its exact value during testing is not a trivial exercise. As explained in answering the next case study question, the deployment and evaluation of non-functional requirements like reliability is a problem noticed by many researchers. This conclusion is therefore assumed to have validity beyond the studied environments.

*7. **To what extent are maintainability requirements defined, deployed and evaluated?***

In all cases, maintainability was mentioned as an important project objective. However, in none of the cases was maintainability defined in quantitative terms.

Only in case G was evidence found that, during the design or architecture phase, time and effort was spent to deploy maintainability in identified components. In case C, very detailed design and coding rules were available with the objective of contributing to high product maintainability.

It may not always be possible to conduct a simple mathematical breakdown of non-functional requirements to low-level components. This is true for most quality attributes as distinguished in the various quality models like the ISO/IEC 9126 standard (ISO 2001b, 2003b, 2003c, 2004e), based on research and publications of, for example, McCall *et al.* (1977) and Boehm *et al.* (1978). This standard distinguishes six different classes of product attributes, like 'maintainability', which are mapped onto quality sub-characteristics, as in Figure 4-12.

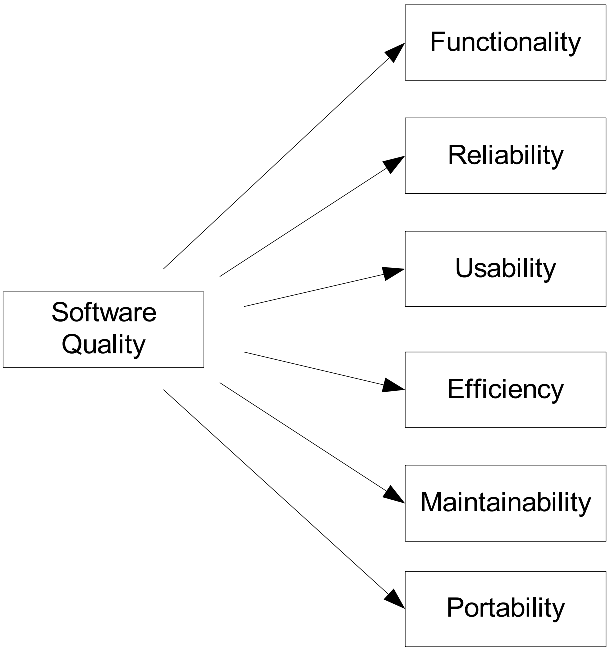

Figure 4-12: Attributes describing Software Quality
(ISO 2001b)

Kitchenham and Pfleeger (1996) consider these quality models share certain common problems:

- ❖ They lack a rationale for determining the hierarchy [characteristics and sub-characteristics in ISO/IEC 9126; factors and criteria in the McCall model], making it impossible to use the model as a reference to define all non-functional requirements.

- There is no description of how the lowest level metrics [indicators in ISO/IEC 9126] can be used to evaluate non-functional requirements at a higher level. In this case implementation rules may be defined that will implicitly contribute to meeting the non-functional requirement at product level. Parnas (1997), for example, describes how a high level of extension, or maintainability, can be obtained through design rules. Dromey (1996) suggests instead a bottom-up approach by defining and building in a consistent, harmonious, complete set of product properties and linking these product properties to the high-level non-functional requirements. He distinguishes the following categories of product property classifications for quality: correctness properties, internal properties, contextual properties and descriptive properties. He identifies two categories of implementation components: those that describe computations and those that describe data. An empirical process is used to determine the quality-carrying properties for each implementation component and their relationship to the property classification(s).

In none of the cases was product maintainability evaluated prior to the release decision. Only in case C was it verified that the detailed design and coding rules were followed, implicitly contributing to high product maintainability. Examples of a poor repository for defects were found in various cases [B, C, D, E], but only in a limited number of cases [C, D, E], were analysis activities conducted to find root causes for defects.

It is concluded that maintainability is a difficult non-functional requirement to define, deploy and evaluate; a conclusion assumed to be valid beyond the studied environment.

8. How do software manufacturers estimate post-release operational cost, for short-term corrective activities and long-term product enhancements, prior to the release decision?

In none of the cases studied was information available to estimate the expected post-release maintenance effort. This observation holds both for the corrective maintenance effort and the adaptive/perfective maintenance effort. Reasons were:

- The reliability level was uncertain, making it difficult to [accurately] estimate the expected number of post-release defects.
- The average effort or cost for correcting a defect was hardly known. Even when the reliability level could be quantified, the corrective maintenance was difficult to calculate.
- The maintainability of the product was basically unknown, making it difficult, if not impossible, to state the extent to which a product can be further adapted, or perfected, in the future and the associated costs.

Another important observation here is that the information regarding the availability of relevant documentation and the quality of this documentation was limited in several cases [A, B, E]. This is expected to undermine the efficiency and effectiveness of correcting defects, or giving the product additional quality, especially when this discrepancy occurs during initial product development (Lientz and Swanson 1981; Krasner 1989; Cook and Visconti 1996).

It is concluded that the estimate of post-release operational cost for short-term corrective activities and long-term product enhancements, prior to the release decision, is a difficult task due to problems in determining exact levels for the reliability level obtained, and the maintainability of the software product. It is assumed that quantifying these non-functional requirements is a general problem in the software industry due to a lack of models that have been sufficiently validated in a practical context. An alternative approach could be the collection of experiences data from past projects, and making these available for use in future projects.

4.4.3 Decision-making Perspective

9. To what extent is a formal collective decision-making process applied to software release decisions?

In all cases but one [A], process descriptions for software product development were found. In practice, these process descriptions were followed in cases B, C, F and G, and partially followed in cases D and E. This observation can be explained using the process capability of the organizations involved. Case C, with the highest process maturity, has the availability of a documented and institutionalized process, whereas case A, with the lowest process maturity, has no documented or institutionalized process. The other organizations are in between these two extremes.

The process descriptions found did not however focus explicitly on software release decisions. No descriptions were found for the implementation process of the release decision. In the cases studied, the software release decision presented itself as a contingent decision, in the sense that the decision to release had, in fact, already been made, but that it had been put on hold until a reliability condition was met. Different release alternatives were not explicitly considered, but testing was continued until the product was 'stable enough'. In all cases studied, the decision-makers implicitly followed the 'good enough' approach, as described in Section 3.3.2, although facing difficulty in making firm statements regarding the achieved reliability level and maintainability in quantitative terms, as drivers for expected post-release maintenance costs.

It is concluded that the cases studied did not reveal process descriptions explicitly focusing on the software release decision-making process. Through the questionnaires and during the interviews, the informants confirmed that no formal decision-making process for release decisions was available, but that their organisation probably would benefit from such a process by creating transparency regarding responsibilities [who], activities [what], timing [when] and supporting methods [how].

10. How can a software release decision be characterized?

In Figure 4-13 an overview is given of the characteristics of the release decision as found in the cases studied. This overview includes the version of the product developed and the perceived strategic value of the decision.

The non-routine [non-recurring] decisions appeared in new or first product developments [A, B, E, G], whereas routine [recurring] decisions were found for subsequent versions of software products [C, D, F]. The perception of strategic value was high for all new product developments [A, B, E, G] as well as for most subsequent versions [C, F]. Only in case D, was the perceived strategic value low. It is assumed here that the characterisation of the release decision is influenced by the phase in the product's life-cycle and perceived strategic value. If a new product with strategic decision value is developed, the cases reveal that the release decision is non-routine [Category II] as many uncertainties are present. For subsequent product versions with low-to-medium and high strategic decision value, the release decision is a routine decision [Category I] as knowledge is available on how previous versions of the product have performed. For new product developments with low-to-medium strategic value, it is not clear whether the release decision will be routine or non-routine; these situations were not found in the cases studied. It can be argued that this depends on the level of uncertainty involved – if the uncertainty level is high, it might be characterized as a non-routine decision; if not, a routine decision, as in Figure 4-14.

These conclusions are important as the decision category determines the organizational level at which a decision is to be made (Harrison 1987, p.23). Higher management should concentrate on non-routine decisions [Category II], and routine decisions [Category I] should be left to operating management. An observation is that in all cases studied, but one [Case A], the stakeholders involved were mainly represented at the tactical level. A release decision will often involve co-ordination between different departments within an organization, thus enforcing decision-making at a tactical level, while another important reason to involve stakeholders at the tactical level is that their high level support for the product, from the development stage to its launch to the market, is found to be a success factor for successful product innovation (Rothwell *et al.* 1974; Maidique and Zirger 1984). However, it was expected to observe active involvement of higher management in those cases where the perceived strategic value of the release decision was high [non-routine: A, B, E, G; routine: C, F]. In these cases, their involvement should not be limited to being a stakeholder in the release decision-making process, but throughout product development from the proposal phase until successful release implementation. In some of the cases studied [B, E], higher management became informed/involved when the release date was postponed several times. A passive attitude was observed whereas pro-active behaviour was expected.

Case	Reason to Invest	Decision	Product Version	Strategic Value
A	Development of new system to increase productivity	Non-routine Complete uncertainty Repetitive Non-compensatory	first	high
B	Strategic investment to enter market with new product	Non-routine Complete uncertainty Repetitive Non-compensatory	first	high
C	Investment to implement enhanced functionality	Routine Complete uncertainty Repetitive Non-compensatory	subsequent	high
D	Investment to implement enhanced functionality	Routine Complete uncertainty Repetitive Non-compensatory	subsequent	low
E	Strategic investment to enter market with new product	Non-routine Complete uncertainty Repetitive Non-compensatory	first	high
F	Investment to implement enhanced functionality	Routine Complete uncertainty Repetitive Non-compensatory	subsequent	high
G	Replacement of existing system to increase productivity	Non-routine Complete uncertainty Repetitive Non-compensatory	first	high

Figure 4-13: Overview of Results

Product Version	Strategic Value	
	Low - Medium	High
First	Category I or Category II	Category II
Subsequent	Category I	Category I

Figure 4-14: Decision Category

The second perspective is the level of uncertainty in the software release decision, on the external environment and available alternatives. In all cases, complete uncertainty is identified as the chance of the occurrence of different scenarios could not [or at least, was not] quantified with probability or possibility values. It is concluded that in a practical context it is difficult, if not impossible, to avoid complete uncertainty. The number of different scenarios – all possible combination of states for the partitions, or criteria, considered – is expected to be enormous for the multitude of external [market environment] and internal [product, organization] factors. See also Figure 4-15 with different sources of uncertainty identified by Browning (1998), who assimilated the risk taxonomies of many authors.

Dimension	Description
Performance risk	Uncertainty in the ability of a design to meet desired quality criteria [on one or more dimensions of merit, including price and timing] and their consequences
Schedule risk	Uncertainty in the ability of a project to develop an acceptable design [i.e. to sufficiently reduce performance risk] within a set time and the consequences
Development cost risk	Uncertainty in the ability of a project to develop an acceptable design [to sufficiently reduce performance risk] within a given budget and consequences
Technology risk	A subset of performance risk: uncertainty in capability of technology to provide performance benefits [within cost and/or schedule expectations] and the consequences
Market risk	Uncertainty in the anticipated utility or value to the market of the chosen 'design to' specifications [including price and timing] and consequences
Business risk	Uncertainty in political, economic, labour, societal, or other factors in the business environment and the consequences

Figure 4-15: Dimensions of Product Development Risks
(Browning 1998)

The third perspective is the number of times the release decision was considered. In all cases studied, the release decision was reconsidered several times. The release decision presented itself as a contingent decision, in the sense that the decision to release had in fact been made already, but that it had been put on hold until the product was 'stable enough'. It is assumed that the decision-makers used strategies or *heuristics* based on past experiences to determine what 'stable enough' means.

The last perspective observed is the way information is processed. It is found that all release decisions are non-compensatory, in the sense that a cut-off point was set on the reliability dimension, and testing was undertaken until the cut-off point was met [although the cut-off point was not clearly defined, and it was not possible to accurately determine whether it was met]. In this sense, a software release decision can also be characterized as a contingent decision.

Conclusions on the case study results and their validity beyond these case studies are, as summarized in Figure 4-16:

❖ *Routine*. A software release decision can be both a routine [Category I] and a non-routine [Category II] decision. It is found that the category applicable primarily depends on the phase in the product's life-cycle and the perceived strategic value of the decision to the organization. Strategic release decisions for first product versions are non-routine decisions, whereas release decisions for subsequent product versions are routine decisions. Low-to-medium strategic release decisions for first product versions can be both routine and non-routine decisions. These conclusions are likely to be true in general. First product versions normally have a high uncertainty level,[50] whereas subsequent versions of existing products have a lower uncertainty level where knowledge becomes available on how previous versions of the product performed. It is considered likely however that a situation can be found where subsequent product versions are released to, for example, highly dynamic markets, increasing the level of uncertainty and thus turning a routine decision into a non-routine decision.

- *Uncertainty*. All release decisions can be characterized as problem solving with complete uncertainty. This conclusion is considered valid for any software release decision with, or without, strategic value, using a similar motivation as above, for the routine of software release decisions. It is practically impossible to reduce the total uncertainty space to a limited number of states and scenarios with the assignment of probability or possibility values to each scenario.
- *Occurrence*. All release decisions were considered more than once and therefore have a repetitive character. This conclusion is valid for all strategic software release decisions with the motivation that the non-routine character of software release decisions and the presence of complete uncertainty will, in a practical context, force decision-makers to consider the release decision more than once. In the case of software release decisions without strategic value, the release decision as a routine decision may not necessarily be repeated and can be both repetitive and non-repetitive.
- *Information processing*. All release decisions are non-compensatory, in the sense that a cut-off point was set on the reliability dimension and testing was undertaken until the cut-off point was met. This conclusion is not necessarily valid for all software release decisions. It might well be the case that, in specific circumstances, like entrance to a new market and/or commercial mass-markets, the trade-off between reliability and time-to-market [first-mover advantage] is considered appropriate.

Characteristic	**Strategic Value**	
	Low - Medium	**High**
Routine	Routine	Non-routine
Uncertainty	Complete Uncertainty	
Occurrence	Repetitive or Non-repetitive	Repetitive
Information processing	Compensatory or Non-compensatory	

Figure 4-16: General Characteristics of Software Release Decisions

11. Which decision-making models apply to software release decisions?

To determine whether the characteristics of software release decisions correspond to one or more decision-making models, the key assumptions of the models presented by Harrison (1987) are evaluated:

[50] One could define the hypothesis that a correlation exists between the level of routine and process capability. Motivation might be the improved control of the development process and the availability of historical data at higher maturity levels, thereby reducing uncertainties. The cases studied did not support this hypothesis. In two cases with higher maturity levels [case E officially assessed at CMM level 3; case G: estimated CMM level of 2-3], a new product was developed but serious problems were encountered. A possible explanation: The development process is only one source of uncertainty; examples of other potential sources for uncertainty are, for example, dynamic market conditions, and introduction of new technologies, in the product itself, as well as in the development environment.

- ***What is the nature of the decision objective[s] [fixed, attainable, limited or highly dynamic]?***
  The nature of the objective of the final release decision turned out to be attainable in the cases where a subsequent version of an existing product was developed [C, D, F]. Deviations from the originally-stated project objectives were for instance: the planned release date was postponed, the functional requirements were decreased, the non-functional requirements were not met, and product and development processes used do not comply with stated standards. In the other cases [A, B, E, G], the objectives were not just attainable but highly dynamic with considerable budget and schedule overruns. This involved the development of the first version of a new product, where in all cases either the infrastructure changed [A, G] or new product technologies were developed and applied [B, E].
- ***Which information is available to make the decision? Is it complete and reliable?***
  In all cases the information available to make the final release decision was limited; neither complete, nor reliable. In all cases, the exact reliability level and product maintainability were unknown. This makes it difficult to accurately predict the operational cost when the product is transferred to its intended users. This was found to be especially true for new products [A, B, E, G]. For subsequent versions of existing products with relatively minor changes [C, D, F], the available information was more complete and reliable [but not perfect] due to the availability of a record of how previous versions of the product had behaved in an operational context.
- ***Does the needed information exceed cognitive limitations?***
  During the case studies it was thought to reveal whether information needed to make the release decision exceeded cognitive limitations. This turned out to be too difficult in the interviews. However, according to present theory, it is assumed that cognitive limitations play a role when a software release decision is made by a group of individual stakeholders. This is further discussed in Section 8.2.1.
- ***Do cost and time constraints exist?***
  In all cases, cost and time constraints were present in retrieving complete and reliable information. Although, in most cases, considerable time and cost was spent on integration and system testing, to detect possible defects and verify functional requirements were correctly implemented, and that a ‘sufficient’ level of reliability had been obtained. The available resources were not unlimited for other aspects.
- ***To what extent can different release alternatives be compared quantitatively?***
  In all cases, release criteria could not all be quantified. In all cases it was difficult to accurately evaluate the reliability level and product maintainability, with a quantitative method, on a ratio scale. It is further difficult to quantify compliance to standards other than on an ordinal scale like: *no – partially – yes*. This makes it difficult, if not impossible, to quantitatively compare different release alternatives.
- ***Is the environment open or closed?***
  In all cases, the environment was open. It is practically impossible to control all possible variables completely as release decision-making is not a contrived situation.
- ***Is the outcome limited in qualitative and quantitative terms?***
  As reliability and product maintainability were not precisely known in all cases, the outcome of the release decision is limited in quantitative terms. It is difficult, if not impossible, to accurately predict the operation cost, for example, for corrective and adaptive/perfective maintenance activities. There are also qualitative limitations. What is the effect on the outcome in qualitative terms, for example, if the internal standards have only been partially met?

The results obtained are presented in Figure 4-17, where ‘+’ denotes a match between a criterion and decision-making model (see Section 3.4.3 for a description of the characteristics of the decision-making models). It shows the key assumptions discussed above [*Objectives – Outcome*] with the addition of a time-horizon. The following conclusions are drawn:

- The Rational Model does not apply as the information is imperfect, cognitive limitations exist, time and cost constraints are valid, alternatives are only partially quantifiable, the environment is open, and the outcome is quantitatively limited.
- The Organizational Model applies in the cases where the objectives are attainable and the time-horizon is short-term. The outcome is qualitatively and quantitatively limited.
- The Political Model does not apply as it aims toward an outcome that is acceptable to many external stakeholders instead of meeting objectives that benefit the organization (Harrison 1987, p.156).
- The Process Model applies in the cases where the objectives are highly dynamic and the time-horizon is long-term. The outcome is qualitatively and quantitatively limited as in the Organizational Model, but a major difference is that the outcome is directed toward meeting highly dynamic objectives such as entering a market with a new product.

Criterion	**Rational [classic]**	**Organizational [neo-classical]**	**Political [adaptive]**	**Process [managerial]**
Attainable/dynamic objectives	–	+	–	+
Imperfect information	–	+	+	+
Cognitive limitations	–	+	+	+
Time and cost constraints	–	+	+	+
Non-quantifiable alternatives	–	+	–	+
Open environment	–	+	+	+
Limited outcome	–	+	–	+
Time-horizon [short, long]	+	+	+	+

Figure 4-17: Results on Decision-making Models
(models described by Harrison 1987)

It is concluded that the characteristics of software release decisions match with the key criteria of two decision-making models: the Organizational Model and the Process Model. Differences are the time-horizon [short versus long] and the nature of the objectives [attainable versus [highly] dynamic]. The Organization Model matches the characteristics of routine software release decisions [attainable objectives and short-term time horizon], whereas the Process Model matches the characteristics of non-routine decisions [highly dynamic objectives and long-term horizon]. This conclusion is assumed to have validity beyond the study environment.

As the scope of this research is strategic software release decisions, being non-routine decisions, the Process Model is used as a reference in this study for the methodology to be designed.

12. Which models of choice are used for software release decisions?

Different release alternatives are not explicitly considered in the case studies. When answering case study question 10, it is observed that all release decisions made in the cases studied were non-compensatory, in the sense that a cut-off point was set on the reliability dimension and testing was undertaken until the cut-off point was met. However, as discussed in case study question 10, this conclusion cannot be generalized.

4.5 Aggregated Results

Most cases studied reveal strategic software release decisions; the existence of prospective large financial loss outcomes, including the presence of high costs for reversing the decision. All release decisions studied reveal complete uncertainty; little quantification with probability or possibility values was possible. As argued in Section 1.5, decisions with a high impact and the presence of high uncertainty can benefit from a formal collective decision-making process.

Aggregating the results of the exploratory case studies leads to four main identified problem areas:

1. *Definition of the release criteria*. Documented and commonly-accepted product development strategies were not common in the cases studied [Section 4.4.1]. Not having consensus among stakeholders about priority setting in a product development strategy could imply that stakeholders do not work towards a common goal. It leaves room for self-imposed controls and restrictions, and performing activities [costs] that add no value.
2. *Information about the implemented values of the release criteria*. In all cases, information as input to the decision-making process was incomplete. Two examples are [Section 4.4.2]:
   - In most cases non-functional requirements were not deployed during product development [design, implementation, and test]. It was only during testing that reliability again received attention, which may be too late to guarantee a high reliability level.[51] The level of maintainability obtained was not addressed.
   - Information on the availability of relevant documentation and the quality of this documentation was limited in a number of cases.

   As a result, organizations faced difficulty in making firm statements about expected post-release maintenance costs.
3. *Decision-making process*. The process descriptions found did not explicitly focus on software release decisions [Section 4.4.3]. Through the questionnaires, and during interviews, informants confirmed that no formal collective decision-making process for release decisions was available, but that their organisation probably would benefit from such a process by creating transparency on responsibilities [who], activities [what], timing [when], and support methods [how].
4. *Implementation of the release decision*. The process descriptions found paid no or limited attention to the implementation of the release decision, once it was made [Section 4.4.3]. Although, in all cases, corrective actions were implemented for defects found after the release decision implementation, most cases revealed the absence of an institutionalized process to analyse the defects found and evaluate the business case, or project, afterwards to supplement organizational knowledge. This makes it difficult to plan expected post-release maintenance costs for future projects based on prior experience, and prevents software manufacturers from identifying areas for improvement.

The problem areas identified in these exploratory case studies corroborate the need for a methodology.

4.6 Adoption Requirements

Another important observation is that existing methods and models for investment evaluations, software cost estimation, architecture evaluation, software defect prediction/estimation, and

[51] Testing can reveal the presence of defects but not their absence. It is impossible to test quality into a product: it should already be there (Deming 1982).

multi-criteria decision-making are hardly known and seldom used in the cases studied. Why are these models not adopted, or only in limited adoption, in the software industry? According to Raghavan and Chand (1989), one reason is that many practitioners are not mature, which is revealed when they ask questions, such as:

- Do experimental results prove conclusively that the practice will work in the field?
- Are successes a result of the practice itself, or might they be the result of the people using it?
- Is the practice complete, or does it need to be adapted or extended before it can be applied?
- Does the practice have significant overhead [training, documentation] that offsets its value in the long run?
- If the practice was developed in a research setting, does it apply to real-world problems?
- Does the practice generally slow down programmers?
- Can the practice be misapplied?
- Is information available about the risks involved in using the practice?
- Does the practice include information on how to integrate it with existing practices?
- Must the practice be applied in its entirety to realize significant benefits?

According to Garland (1991, p.283), major barriers to be considered in diffusion and adoption of an innovation are people issues, including cultural traditions, risk aversion, lack of knowledge and user acceptance. Rogers (1962) defines diffusion as the process by which an innovation is communicated through certain channels, over time, among the members of a social system. Using this definition, he distinguishes four elements of diffusion (Rogers 1962):

- *Innovation*. According to Rogers (1962), an innovation is an idea, practice, or object that is perceived as new by an individual or other unit of adoption. Five characteristics of innovations are identified to help explain the differences in adoption rates (Rogers 1962):
  - Trialability: can be tried on a limited basis before adoption.
  - Observability: offers observable results.
  - Relative Advantage: has an advantage relative to other innovations or the *status quo*.
  - Complexity: the methodology is not overy complex.
  - Compatibility: is compatible with existing practices and values.
- *Time*. Time relates to the speed with which an innovation is adopted by potential adopters. The theory holds that an innovation will experience an increased rate of diffusion if potential adopters perceive that the innovation is positively related to perceived relative advantage, compatibility, trialability, and observability, but is negatively related to perceived complexity.
- *Communication*. Rogers (1962) states that diffusion is a particular type of communication in which the message content exchanged is concerned with a new idea. Thus, the diffusion process involves the spread of a new idea from its source to potential adopters.
- *Social system*. Rogers (1962) defines a social system as a set of inter-related units engaged in joint problem-solving to accomplish a common goal. Put differently, the members of a social system may be individuals, informal groups or organizations who work toward the achievement of a common [set of] goal[s]. He further states that the culture within the social system and the individuals who make up the social system can affect the diffusion of new ideas.

Rogers' Innovation Decision Process theory states that diffusion is a process that occurs over time and can be seen as having five distinct stages (Rogers 1962):

- *Knowledge*: exposure to its existence, and understanding of its function.
- *Persuasion*: the forming of a favourable attitude.

- *Decision*: commitment to its adoption.
- *Implementation*: putting it into practice.
- *Confirmation*: reinforcement based on positive outcomes.

According to this theory, potential adopters of an innovation should first learn about the innovation, be persuaded as to the merits of the innovation, decide to adopt it, implement the innovation, and confirm [reaffirm or reject] the decision to adopt the innovation. While Sachs (1993) correctly concludes that many other important theories of innovation diffusion are overlooked, the Innovation Decision Process theory remains among the most useful and well known.

Based on the case study findings (see Section 4.4.2), it is concluded that the software manufacturers involved are mostly in the first stage of the Innovation Decision Process for many models. They are being exposed to the existence of these models. This situation may well be true for the software industry in general: both software engineering specific models [cost estimation, architecture evaluation, defect estimation and prediction] and models originating from other disciplines [e.g. economics, social psychology] are relatively unknown and unused. One could argue here that the software industry is a relatively young discipline and that specific models have not been sufficiently validated for their practical applicability. This argument is however only partially true. A software cost estimation method like COCOMO, for example, was initially developed in the early eighties and was based on empirical data.[52]

For the methodology to be designed, it is important to understand that the rate of adoption is positively related to perceived relative advantage, compatibility, trialability and observability, and is negatively related to perceived complexity of such a methodology.

4.7 Summary and Conclusions

In this Chapter results of the case studies are presented, with answers to questions raised in Chapter 3.

For the economic perspective, it is concluded that the availability of formulated, documented and accepted product development strategies is not common in the cases studied. In practice, reliability of the product is an important issue prior to the release decision, whether formulated as such in the initial product development strategy or not. This was true both for new versions of existing products and first versions of new products. This means that Moore's model on shifting product development strategies could not be validated in the cases studied; it depends heavily on factors like the characteristics of the market and the reputation and competitive positions of a manufacturer organization. A purely economic cost-benefit analysis to make a computational release decision is found to be difficult if not impossible: not all parameters can be quantified with sufficient certainty and the exact relationships between all parameters are unlikely to be known in a practical context. It is assumed that this conclusion is valid beyond the studied environment. The market entry strategy chosen in the environments studied was to continue testing until reliability achieved was considered 'good enough'.

From the software management perspective, the case studies reveal that estimates for development cost and schedule were derived using experts. Reliability and maintainability were important non-functional requirements in the cases studied, but it is concluded that the deployment of their defined values to design components as well as the evaluation of their exact values during testing is not a trivial exercise, as there is no description of how lower level

[52] COCOMO, later renamed to COCOMO 81, was published in 1981 (Boehm 1981), COCOMO II is its successor (Boehm *et al.* 2001).

metrics can be used to evaluate non-functional requirements at a higher level. This conclusion is assumed to be valid beyond the studied environment. In the cases studied only reliability was evaluated [whether defined as important or not], whereas product maintainability was not explicitly addressed [whether defined as important or not]. However, the evaluated reliability level was uncertain, making it difficult to [accurately] estimate the expected number of post-release defects, while the average effort, or cost, for correcting a defect was, basically, unknown. So, even when the reliability level could be quantified, the corrective maintenance is difficult to calculate. The maintainability of the product is unknown, making it hard, if not impossible, to make statements about the extent to which the product can be further adapted, or perfected, in the future, and against what costs. Finally, the availability of relevant documentation and the quality of this documentation was found, in several cases, to be limited. This undermines the efficiency and effectiveness of correcting defects, or adding additional functionality, especially when this occurs during initial product development.

On the decision-making perspective it is concluded that a software release decision can be both a routine and a non-routine decision. Release decisions for subsequent product versions with low-to-medium strategic value will normally be characterized as routine decisions and match the criteria of the Organizational Model [attainable objectives, short-term horizon], whereas the first version of a product with high strategic value will normally be characterized as non-routine decisions and match the criteria of the Process Model [dynamic objectives, long-term horizon].

Other characteristics of software release decisions are the presence of complete uncertainty, its repetitive character for release decisions with strategic value [not necessarily true for release decisions without strategic value], and the way information is processed: with or without compensation. These characteristics of software release decisions are assumed to have validity beyond the studied environment. In the cases studied, the software release decision presented itself as a contingent decision in the sense that the decision to release had, in fact, already been made during the test phase, but that it had been put on hold until some cut-off point for reliability was met. This cut-off point could however not be accurately determined. In the cases where the strategic value of the release decision was high, pro-active involvement of higher management was expected throughout product development. However, the involvement of higher management was limited to a passive attitude when it became obvious that the initial project objectives could no longer be met; while they were informed about this, corrective actions were left to management at the tactical level.

Most cases studied revealed strategic software release decisions: and the existence of large prospective financial loss outcomes, including the presence of high costs for reversing decisions. Furthermore, all release decisions studied reveal complete uncertainty: only limited quantification with probability or possibility values was possible. In the cases studied, no explicit descriptions were found of a formal decision-making process for software release decisions. The informants stated that their organisation would probably benefit from such a process by creating transparency of responsibilities, activities, timing, and supporting methods for strategic software release decision. The need for a methodology was thus confirmed in these exploratory case studies; decisions with a high impact and the presence of high uncertainty require a formal collective decision-making process.

A final observation on the cases is that existing methods and models for investment evaluations, software cost estimation, architecture evaluation, software defect prediction/estimation, and multi-criteria decisions are hardly known, and seldom used. This gap has been identified before, for example, by Poore (2004) stating that software engineering has suffered a decided lack of direction. Using the Innovation Decision Process theory, the software industry in general seems to be in the first stage of exposure to the existence of these models. For the methodology to be designed, it is important to understand that the rate of adoption is positively related to perceived relative advantage, compatibility, trialability and observability, and is negatively related to perceived complexity of such a methodology.