Universiteit Leiden

ICT in Business

An integrated multi-faceted system approach for improving the development of information systems

Well begun is half done!

Name: Antonia Wildvank
Student-no: 1587536

Date: 30/07/2019

1st supervisor : Dr. H.T.(Hans) Le Fever

2nd supervisor: Dr. A.J. (Arno) Knobbe

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Leiden University

Niels Bohrweg 1

2333 CA Leiden

The Netherlands

An integrated multi-faceted system approach
for improving
the development of information systems

Well begun is half done!

Mimmo Paladino, Untitled 2015

Master's Thesis

Antonia Wildvank

SUMMARY

Reason and problem statement

Organisations increasingly use Information Technology with the intent to improve their efficiency or effectiveness. The introduction of Information Technology is usually conducted in the form of one or more projects under the governance of a project board. The design, construction and implementation activities of Information Technology are usually well defined and described in the Information Technology domain. Therefore, delivering functionally and technically correct products (IT systems) should be achievable. However, while the introduction of new technology is intended to impact the functioning of the whole organisation, the outcome of that activity seems less effective as many IS/IT-projects fail to deliver results. The Standish report from 2015 shows that this problem is not typical Dutch (Standish, Chaos report 2015): only 29% of the projects in the global IT sector showed to be successful, and 19% is a complete write-off. The Dutch Bureau of ICT Reviews (BIT) concluded for only 5 out of 50 projects that they probably would succeed without essential adjustments.

However, project boards try to diminish the risk of failure by requesting project audits or evaluations. They steer based on intermediate results from the project team in which they would have to apply various criteria such as quality demands and path dependencies. However, Bronsgeest (2016) has shown that over 95% of the evaluations (114 out of 117) focuses only on the process followed. This outcome led to the hypothesis underlying this research that project boards do not take the quality of deliverables much into account.

The costs of fixing errors caused at the start of projects vary according to the progress in the systems development cycle and vary from 1, 6.5, 15 to 80 times the time already used, when they have to be fixed in the requirements, coding, test or implementation (Pressman, 2005). Intervention at an early point helps, but preventing these errors is the attempt of this research by providing a model for the Project Boards to improve their steering.

Therefore, the following research question has been formulated:

> *Which deliverables or documentation (artefacts) of systems development projects are crucial for clients to steer upon, to get a higher chance of a good end result of a project?*

Also, the related design-objective is formulated as:

> *Also, if these can be found, is it possible to construct a model that could support Project boards in steering the project?*

Method of working

In order to achieve the research goals, and thereby answering the research questions, a research design was constructed based on Design Science in Information Systems research (Hevner, 2004).

- The knowledge base was established by a literature review, which was based on the nouns and concepts in the research question.
- The conceptual research model was built and justified by
    - a field study (survey) to investigate relations/differences between successful and unsuccessful projects.
    - Two case studies, to analyse two unsuccessful projects.

The literature review provided insight into the nature of systems in general and the interdependencies between different (layers of) systems. These levels were projected on the more or less implicit system concepts in systems development theories (especially software development theories) and on contracting, project management methodologies and their artefacts.

The insights on the comparison of methodologies and systems levels led to the construction of the first conceptual model.

This conceptual model was used to evaluate two case studies. These two case studies were purposely chosen because of their entirely different substance. One (Private) was the refactoring and extension of a package, where supplier and client both were commercial companies. The other (Public) was based on an agreement between two entities (supplier and client both in the public domain) and included the development of custom-made software.

To evaluate and detail the model with the relevant business needs, a survey was conducted amongst decision makers in project boards (of different projects) to understand if there is a difference between well-run projects and failed projects. In total, 38 completed the survey. The following hypotheses for the successful completion of a program/project were tested:

- Successful programs/projects have, as a driver for the business case, a clear goal that can be described by the business owner
- The constituent elements of the business case are known
- To deliver the business case processes are known/developed before programming and alternatives are considered.
- In the case of customised development, ISO25010 norms are part of the requirements
- Quality management is present and positioned outside the program/project (for the business owner to get a clear view of the results).
- Methodology matters and relations between methodology and activities are essential.
- Each type of contract needs its specific preparation up front.

Findings

The literature study showed that general systems theories are defining systems by distinguishing layers in systems: supra-systems, systems and subsystems. The IT system is one of the lowest in that hierarchy. It is serving processes and functions (an Information System) which serves people (another "subsystem") to fulfil their role in line with the purpose of their organisation (the layer system in general systems theories) and thus causes an effect in the outside world (clients, citizens, partners, other organisations: the supra-system).

Systems theory shows that systems have their equilibrium and tend to remain as they are. The purpose of a project is to deliver a sustainable change. To reach the desired goal, all layers of a system have to be considered, as well as specific measures.

The most important findings of this literature study on software development methodologies (SDM's) and related methodologies are:

- In many methodologies, the definition of systems is lacking.
- SDM's do not address the same levels of the general systems theories.
- Contract levels address the same levels of systems as in general theories.
- SDM's do not cover the same activities of the ISO12207 development standards.

- The initiation, realisation and transition phase of SDM's do not contain the same activities.
- An SDM alone is not sufficient to deliver the result: other systems development methodologies are necessary as well.

Both **case studies** showed, although seeming entirely different, that Client and Supplier failed to understand that they did not get a joint agreement about the way to reach the end result. This caused ambiguity in contracting by not addressing the essential parts of the general SDM-activities as described in ISO12207. In both cases, the realisation phase started before these questions were answered. Following the list of hypotheses stated earlier, the only one that appeared to be true is that each type of contract needs its specific preparation. The specific methodology did not matter; instead, the absence of some artefacts of ISO12207 mattered. Quality management was in place after the preparation phase, but the problems occurred beforehand.

For the **survey,** 38 project board members were asked to take in mind a completed project. However: many projects were not completed. This discrepancy could demonstrate a “governors’ bias”, i.e. their belief results would be met, even when missing elements in the process would indicate otherwise.

The results of the survey showed:

- Former methodologies such as RUP, DSDM (also Agile) are hardly used anymore; Agile/Scrum is said to be chosen explicitly (67%) by project boards.
- Many of the projects and programs are not contracted but delivered internally, sometimes with a hired professional as program manager, without explicitly allocating or contracting responsibility. However, those projects that have been executed as fixed price projects (i.e. with singular responsibility) all deliver within budget with similar results (even slightly better) as the projects managed by the organisation tself.
- Defining goals and considering alternatives before starting a project seem crucial for successful delivery and also causes that projects run within budget.
- The transformation of ISO25010 requirements did not seem crucial in the projects.

The crucial substance of deliverables

Methodology and deliverables are essential in the way that some methods do not cover the whole systems development cycle, especially not those of the preparation or plan phase. It should not only be clear who is responsible for managing the outcomes at each level of the system, but the SDM-aspects that belong to the higher contract levels should be carried out as well to obtain the right agreement. This "mutual or reciprocal agreement” can only be drawn up at the end of the preliminary phase, but this phase is containing more or fewer aspects depending on the contract type and methodology.

The mutual understanding of these preliminary documents is essential as they form the skeleton of the project. However, also important is the mutual understanding and agreement on who is performing what during the realisation phase and the hand-over to the organisation. This "division of work" has to be decided in the preliminary phase since the contract and project plan are based on how this is assigned. These findings from the case studies and the survey have subsequently been incorporated in the Qube (Quality of Understanding the Business Environment) model.

The Qube-model as a result

The ultimate goal of this research was to construct a model as a guideline for clients who have to steer a project. This model (Qube) provides project boards with a model that

- addresses all levels of the system for which the project delivers a result;
- compares the contract-level with the wished outcome
- takes all system-development methodologies into account.

The system-level and contract-level have proved to be the most critical parameters, whereas the activities from the general software development cycle according to ISO12207 have to bridge the gap between what is needed and what is already available.

The Qube model is designed to deliver a mutual agreement, i.e. a document that is the heart of the project plan. It is based on the five levels of the system definition. The project board can quickly assess (or let others assess) whether all preliminary actions have been taken and if not, urge to complete the requirements.

This model is used in practice, and as an extra check the results of the Audit Bureau of Dutch Government, BIT (Bureau of ICT Toetsing), are compared with this model, which led to a few smaller adjustments and a reconsideration of the ISO25010-questions.

Further research

When pursuing this research, many questions and topics arose due to specific findings. In the last chapter, these subjects are touched upon briefly as time constraints leave the following for future research:

1. The professional and ethical standards of suppliers.
2. The public tendering processes in relation to this model.
3. Lean/Agile methods and the EFQM/Rhineland model.
4. The "governors' bias."
5. Process Design, why not?
6. Learning from successful projects.

Reflection

This research led to another outcome than expected as it was assumed that especially non-functional requirements were neglected at the start of projects. The two cases, that were carefully selected for their dissimilarities, turned out to have many similarities at other points, especially in not pursuing the initial activities of systems development right and ill understanding of the methodology used. The results of the survey showed similar results as the case-studies. It was already known that conducting a preliminary phase well is essential but applying this prelimary phase to all faceted levels of a system gave a new perspective.

In that respect, the Qube model is helpful. It has not been tested in a large environment yet, but for the past two years, it has been successfully applied in four projects. Two were started and satisfactorily delivered within a year. The other two derailed before using Qube and got back on track again by using it and subsequently also delivered within a year.

Assisting board members to use this model (and actually the theory behind it) t is the next challenge for the upcoming years.

Index

Figures and Tables

Preface

This thesis serves as the final assignment of the Master Business in ICT of the Universiteit Leiden (Leiden University) of Liacs.

Some people say: “the journey is more important than the goal”. If that is true, the journey was very interesting. Sometimes rough, sometimes very interesting, especially by reading literature again or new and sometimes wondering why projects were heading somewhere else than expected even if thorough knowledge is available. It sometimes made me wonder whether this branch of IT science (methodology and delivering business projects) needs to be developed further or that the dissemination of what is already available needs more emphasis.

After completing this research my humble opinion is that new methodologies could use scientific underpinning more, not neglecting what is already available.

This journey started with the mission my friend Harry Stroeken gave me, who “forced” me to get my master, and was continued with the help of my friends, Ruth and Jaap de Wit, who took care of me when work and study sometimes became the only “art of living”. All my friends and family who put the pressure on me in succeeding and other the other hand accepted my absence.

I thank Hans Le Fever who just let me go my way, trusting that it will be fine by itself. And of course CAI, where I started my journey and LIACS where it will end.

Last but not least I want to thank my colleagues and clients who gave me the inspiration for this thesis.

And, I have a little hope that the goal will contribute a little to a better performance of projects.

1 Introduction

1.1 Software is ubiquitous; projects fail, why?

This research started out of curiosity about the following: New products have conquered our society, often consisting of software alone. In those products (smartphones, PET-Scanners, radiotherapy devices, automatic cars and other products) software is functioning excellent and precise, although some problems occur now and then (but hardly in functioning Radio Therapy Devices). However, many "projects with a digital component", as the Netherlands Court of Audit calls them, tend to be far more expensive than expected, often not delivering results or much later than expected. Many people point to those products and wonder why the same results are not met in "the other" projects.

An essential difference between the two types of software-intensive projects is that in the first, the software is embedded in a product, whereas in the other the software is embedded in an organisation with people. Furthermore, the latter has to cooperate with legacy software in the same system, whereas embedded software might communicate with other systems, but outside its embedding system. Therefore, we call the latter projects Business/IT (or short B/IT) Projects. We discuss the B/IT projects in this thesis and not the product-embedded projects. What exactly is preventing the B/IT projects from being more successful? With that knowledge, we could contribute to improve the success rate.

Most projects have project boards in which stakeholders are represented: the business owner, the supplier, the user representative and sometimes a quality assurance expert. An unsuccessful project did not only deliver less or no results because the project as such failed: the project board was also not able to prevent the project from failing.

This research is not about the situation of intentionally misinformed project boards by withholding information on the results. However, is a project board and especially the users and business owner always able to understand the impact of a decision in a project? In several cases where clients disputed how the supplier handled a project, the Brinkers-arrest (Hoge Raad, 11 April 1986) concluded very clearly: even if the client gives a wrong order, the supplier must act in good faith and when the client makes this impossible, cancel the contract.

In this research, we all call projects and programs "Projects" since the essence of our research applies to both. We will call clients, commissioners, problem owners, business owners, senior responsible owners "Clients" and the contractor, whether internal or external "Supplier". We acknowledge the difference between application and infrastructure systems, but we will use "IT projects" to define both.

The goal of this research is twofold; the first goal is to gain insight which knowledge, included in deliverables is needed to steer a project towards the intended results. The second objective is to construct a model, based on this insight, to support clients in steering a project on substance as well.

In this chapter the context of the problem (1.2) is shown, the problem statement itself is discussed (1.3), its relevance (1.4) and the content structure of the thesis itself is introduced (1.5).

1.2 Research Context

In the evaluations of the Elias Parliamentary Inquiry Committee (Elias, 2015) many B/IT- Projects with lacking results were discussed. Since then, the Bureau of ICT review was installed (BIT). They reviewed until now 51 projects of which at least 45 are judged as incapable to succeed (BIT, 2019). The total amount of money concerning all projects is about 2,2 billion euro (https://www.rijksictdashboard.nl/). In governmental organisations, all failures are reported publicly, but in industries and trade, the situation is not different according to the Standish reports: only 29% of the projects in the IT sector areseen as successful, and 19% is a complete write-off (Standish, CHAOS-reports, 2015).

Bronsgeest (2016) researched what the impact was of evaluations that were executed in the public sector and how the evaluations influenced the results: his findings were that getting a review done was more important than the learning experience. Noticeable was also that over 95% of the evaluations (114 of 117) only focussed on the decision-making process and not on the content quality.

When searching for literature about the relationship on the content of the systems development process and the project management only two recent books were found that cover the subject and even then, only partly (Schwalbe, 2015 and Marchewka, 2014). Either governance in project management is elaborated on, or systems development. The combination of both is thus seldom found.

1.3 Problem Analysis

1.3.1 Motivation

During my career, I was not only involved in professionalising professionals and companies to deliver quality in IT solutions, but also delivered many projects to clients. I recognised that delivering quality is not only about honesty and transparency on both sides, but also concerns the knowledge of clients of what might be expected from the supplier and what they have to provide themselves. In that twilight zone, the negotiations about additional work took place, and the seeds were sown for dissatisfaction with the delivery result. Worse, when there was no discussion at all, and the project derailed, both parties in the best case or only the client were unaware of what had happened.

When projects failed, discussions with project team members often showed that methodologies that were agreed upon were not used and that the project manager took a different direction together with, or ordered by, the client.

These experiences are not enough to draw conclusions from, but it was the beginning of curiosity that ended up in this research study.

The research started with some (at first sometimes unconscious) informal hypotheses based on literature and experience:

- Successful programs/projects have, as a driver for the business case a clear goal that can be described by the business owner
- The constituent elements of the business case are known
- To deliver the business case processes are known/developed before programming and alternatives are considered
- In the case of customised development, ISO25010 norms are part of the requirements
- Quality management is present and positioned outside the program/project (in order for the business owner to get a clear view of the results).
- Methodology matters and relations between methodology, artefacts and activities are essential.
- Each type of contract needs its specific preparation up front.
- Projects are not delivering results because clients lack knowledge about IT.

1.3.2 Relevance

Many studies have been conducted on the topic of failing B/IT-projects. The evaluations on those projects mainly concern the quality of the process and reporting in project management but avoided the discussion whether the client was able to prevent the situation and what the client needed to avoid failure of the project. In steering committees, the interest in the content of the project is often lacking (Bronsgeest, 2016) and steering is mainly focussed on the process.

In this research we deliberately do not take into account power differences, conflicts of interest, incompetence, consciously letting fail the project in the client's organisation nor deliberate underperforming or withholding information by the supplier. We know these things happen (Rooks, 2002), but we do not consider it here and now. For the construction of the model, it seems less critical. The model is meant to prevent these situations by providing knowledge.

As stated above, the Elias Parliamentary commission report (2015) and Bronsgeest (2016) show that the attention of failing projects is more inclined to criticise the process followed than the quality (presence and content) of deliverables of the systems development process.

That is why we use a different line of research to construct a model that might support project boards to get more grip on achieving impact from the project.

1.4 Research question and design objective

The design objective is to construct a model that can predict in an early phase that a project is veering off track and that gives guidance on how to adjust the project.

When developing a system, many types of artefacts (or deliverables or documentation) are produced: analysis, requirements specifications, architectures, designs as the most important ones. Project boards usually do not review them; the quality assurance authority should fulfil that role and give enough insight to the board to adequately fulfil their governance role. The approach in this study is to find out which discussions on which artefacts could improve the result of the project.

Therefore, the research question is:

> *Which deliverables or documentation (artefacts) of systems development projects are crucial for clients to steer upon, to get a higher chance of a good end result of a project?*

And the related design-objective:

> *And if these can be found, is it possible to construct a model to support project boards/clients in steering the project?*

The research question leads to these sub-questions:

- How is a system defined?
- What is systems development?
- What is the definition of a project?
- Which types of artefacts can be distinguished?
- Which artefacts seem crucial?
- How is the end result described?
- What do project boards/clients steer/what question should/could the client ask? And
- Can a model/questionnaire be derived for the client/project board?

Of course, there are many other aspects why projects could fail or succeed: The psycho-social and power context of projects is deliberately not considered. Of course, we are not closing our eyes for it, but these studies belong to another discipline.

1.5 Structure of this thesis

This paragraph contains a short overview of the structure of this thesis. The structure is based on the process of the Design Science Framework (Hevner et al. 2004) which is used to design a new artefact in the realm of IT Projects. We aim at building a model for project boards and evaluate it.

The structure of the thesis follows this method:

- In this first chapter, the main research problem is identified, and the relevance of the problem is discussed.
- In Chapter 2 the research methodology (Design Science) is described.
- Chapter 3 describes the knowledge base for this thesis. It consists of the definitions of systems, the description of all constituent methodologies to deliver a system change and the artefacts used in the of those methodologies, based on a literature study combined with international standards (ISO) for software engineering and project management.
- Chapter 4 contains the first conceptual model and this model is evaluated with two case studies of failed projects and by a survey of successful and failed projects.
- Chapter 5 contains the construction of the detailed model based on the evaluation and justified with the results of the audits of BIT.
- Chapter 6 summarises the results, evaluates the research questions, and suggests topics for further research and gives some recommendations.

2 Methodology

2.1 Design Science

To answer the research question and to construct the model, we used the principles of 'Design Science in Information Systems Research' (Hevner at al. 2004). The fundamental principle is that knowledge and understanding of a design problem and its solution require the building and application of an artefact.

Hevner et al. designed a structured model to show the interdependence between organisational design and the translation of strategy into infrastructure.

Their model contains two main elements: processes and artefacts. The processes are: build and evaluate (at each level of the process) the constructed model and the artefacts could be constructs, models, methods and instantiations.

Hevner et al. discusses IS research methods and positions "Behavioral Science" as a research method that explains or predicts phenomena related to the identified business need. Design Science is the science that addresses research through the building and evaluation of artefacts to meet the business need. They argue that behavioural science is about truth, whereas design science is about utility and that those are inseparable.

An essential distinction in their theory is the difference with "routine design" or systems building, in which organisational problems are solved, whereas the design science as a problem-solving process contributes to methodologies.

Design Science uses seven guidelines to construct this scientifically. These seven guidelines are described below. A description of the application of this guideline in this specific research supplements each guideline:

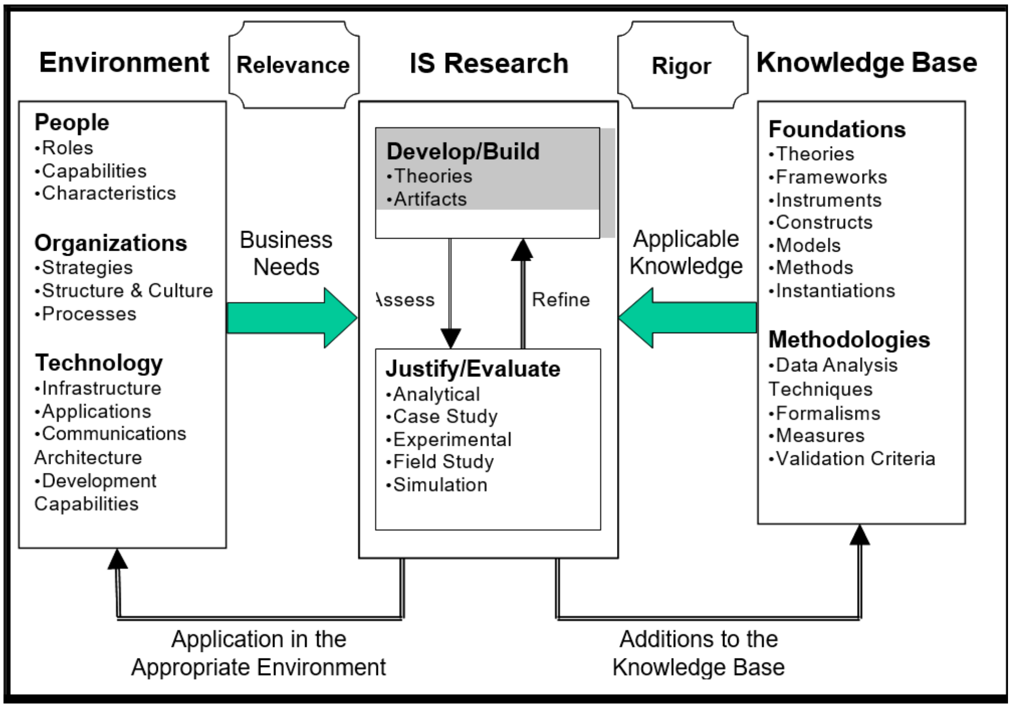


figure 1

Guideline 1: Creation of an innovative, purposeful Artefact

The artefact has to solve an organisational problem: the failure of project boards to steer their projects is such an organisational problem. The purpose is to prevent project boards from failing or at least diminish the rate of failed projects. This artefact is the constructed model: a guideline for project boards and clients. The proof that this artefact leads to this purpose is work in progress. Two projects have been accomplished on time and budget by applying the first models and the model is still applied and evaluated according to the Design method. Further evaluation is added to judge if it could have worked by applying it to the audits of the Bureau of ICT Auditing (BIT).

Guideline 2: Problem Relevance

As stated before: the amount of money lost on failed public projects alone is estimated between 1 and 5 billion euro (Elias, 2015, p 9) in the Netherlands alone and the Standish reports show the same globally, including private projects.

Guideline 3: Design Evaluation

The model has been evaluated three times and in different ways: Minger (2001) promotes the use of multiple methods to "provoke more responses from the world" to enrich the results of the research and make it more reliable. In this research, the model is compared with two case studies of a failed project and by the use of a survey, completed by 38 persons on their projects and afterwards by comparing the results of the BIT-audits.

Guideline 4: Research Contributions

The research must provide transparent and verifiable contributions: the documentation of the projects is present as well as the outcome of the survey. In the next paragraphs the literature review, the evaluation of the projects and the survey design are described. The outcome can be added to the methodology of the technology information realm (like the Plan phase of IT4IT (website 22).

Guideline 5: Research Rigour

In design science, the result is derived from the effective use of the knowledge base (theoretical foundations and research methodologies). The theoretical foundations are based upon literature and described system development and project methods. The testing of the model is done threefold: by comparing the outcome of the survey with the model, by using the cases against the model and by comparing the outcome of the BIT-audits with the model. It is also already used in practice, but details cannot be provided for confidentiality reasons.

Guideline 6: Design As a Search Process

The process is iterative. Problem-solving uses available means, to reach desired ends, while satisfying laws, existing in the environment: in this thesis, we show the iterative process by enhancing the model after each test. In this case, one of the original assumptions was that ISO25010 was not taken into account. After studying the results of the survey, this proved unsustainable. Further study could prove a relation, but this would lead to another research study.

Guideline 7: Communication of Research

The research must be presented both to technology-oriented as well as management-oriented audiences. This thesis report is meant primarily for management, covering an ICT-subject. The intent is that they easily can understand the model. Therefore, it is constructed in business language, based on IT.

The paradigm design science is part of the IS discipline where it gains “further knowledge that aids in the productive application of information technology to human organisations and their management” (ISR, 2002). “It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, management and use of information systems can be effectively and efficiently be accomplished (Denning 1997).

2.2 Literature study (The knowledge base)

2.2.1 Goals

The knowledge base was created by a literature study to create the foundations.

In the first place, the goal was to draw up definitions of the concepts used in this thesis. These definitions are the logical basis for the construction of the model. Therefore, the literature on various aspects was gathered on the specific sub-questions

- Systems (how is a system defined)
- Software development and software development methods
- Projects and project management methodologies (definition of a project)
- Organisations and organisation development methodologies and
- Contracts

In the second-place, literature was used to study success- and fail-factors of ICT projects.

2.2.2 Selection

Scientific literature has been gathered in many ways:

- By reading articles and theses on the aspects as mentioned above (using keywords such as system, contract, software development, methodology) and start from their literature lists
- By reading the report "Grip op ICT" and viewing the many videos of the parliamentary interviews (website 24)) and reading the literature of the interviewees
- By using literature databases and searching on the aspects (using the topics above) and using recent literature, using their literature lists as well.

All literature is selected from the top 25 journals (IS Journal ranking, Saunders, 2008).

Literature concerning methodologies is selected by using the original books of the founder of the methodology or the institution that issues the methodology. Sometimes these are only found on websites. Then they are mentioned as a website on the literature list.

Other literature, not scientific but containing specific information on the subject, is diverse

- The report of the parliamentary interview known as the Elias commission.
- Reports and minutes of the parliament concerning the Public project.
- Websites with information of Dutch Government.

2.2.3 Discussion

The literature review focused on obtaining definitions. Therefore, certain aspects of articles are not discussed since they were not relevant for the goal of obtaining definitions of the concept "system". The extensive literature on failed projects is used in the introductory chapters to underline and expose the problem statement from which the research question derives.

2.3 Design Evaluation

The design evaluation exists of two activities: the study of two projects and the conduct of a survey.

2.3.1 Evaluation of 2 projects

The goal of the study of the two projects was first to find out which similarities occurred in the execution of the projects in different circumstances and secondly to test if the model could explain the results of these projects and if necessary, adjust the model. Therefore, two projects were selected of which extensive documentation was available but were different in relevant aspects.

	Project Commercial	Project Public
Domain	Private	Public
Supplier	Software Company	ICT Department of Public entity
Methodology	Prescriptive/RUP	Agile/Scrum
Contract/Agreements	Capacity	A Project Plan
Intended Result	Functioning Application Package	New and changed chains according to changed legislation.

Table 1

The documents of both projects are not made public but are available for the supervisors of this thesis.

These projects have been analysed using the first version of the model, a conceptual model that is constructed in chapter 3.

2.3.2 Survey

2.3.2.1 *Design*

The survey (Appendix 3) served two objectives:

- to find out in what way successful projects differ from unsuccessful projects, concerning the adjusted model
- to find out if the members of a project board of successful projects perform differently from their colleagues with the less successful projects and if so, in what aspects?

Via SurveyMonkey, the survey was accessible and completed from 11-May-2018 until 8-July-2018.

The survey contained 21 multiple-choice questions, sometimes with the possibility to explain or to give a not prescribed answer. 38 persons out of 100 answered all questions: The survey was spread via e-mail and LinkedIn personal messaging especially to senior responsible owners as Prince2 calls them (business owners, problem owners) and other project board members.

The survey contains questions on the following topics

- the organisation, the role of the answerer and the project (1, 3, 4, 8, 9, 21),
- contracting, costs and governance (4, 10, 11, 13, 18),
- the results and the business case (3, 4, 5, 6, 7, 13)
- questions about methodology (14, 15, 16, 18, 19, 20)
- quality and content (12, 14, 15, 16, 17, 18, 19, 20)

The intention was to see what business owners and others who are accountable for the results of projects and programs know about their projects. The coherence between those answers in a methodological way is more important than the outcome of the projects. However, it is examined if there are any relations.

In this survey the following conditions for a project to be concluded successfully were tested:

- To have a clear goal as a driver for the business case that can be described by the business owner (6, 7, 8)
- The constituent elements of the business case are known (4, 5, 6, 7, 8, 13)

- To deliver the business case processes should be known/developed before programming and alternatives should be considered (14, 15, 16)
- For customised development ISO25010 norms are considered beforehand to deliver quality at once to control costs, and this leads to the intended result (17)
- Quality management should be present and positioned outside the program/project in order for the business owner to have a clear view of the results (12)
- Methodology matters and relations between methodology and activities are essential (2, 14, 15, 16, 17, 18, 19, 20)
- Each type of contract needs its preparation up front (4, 6, 15, 17, 20, 21/28)

2.4 Design, evaluation and testing of the model

The model is meant as a reference card for project board members, to check if all required actions to accomplish the new or renewed system by executing the project, have been taken.

The model has been constructed in three iterations:

First, a conceptual model was designed by combining the logic of the definitions of systems with the coverage of methodologies and contracting/project plans on these concepts.

This model was evaluated and adjusted by comparing the model with the outcome of two failed projects: could the model explain these outcomes? Were certain aspects more or less important?

At last, the new insights were taken into account to design the survey, and after receiving the answers, the model was re-evaluated. Refinements were made, and the model was made ready to use.

The chosen model has also been evaluated by comparing it to the audits the Dutch BIT-organisations performed.

The author tested the model in real life at the start-up of projects. Two of them are already delivered as expected in less than a year. The others are still in progress. The intention is to use this model as a training tool for clients and to be improved further.

3 The knowledge base, literature study

3.1 Definitions, methods and results

Software-intensive projects for products differ from software-intensive projects in organisations as the way the software is embedded, is different: in products the software is embedded in a static product. The product design takes place, and the requirements for the needed software are specified. In organisations, the software is embedded in a dynamic system, which changes all the time. Such a system is a complex of organisational units, people, procedures, politics, interactions with the outside world. Project boards have to deal with this changing environment.

Defining words and concepts is essential to understand each other and be sure that the same topic is addressed. The verb “define” means literally: setting borders (from the Latin “finis”). When Client and Supplier discuss a project and the changes in the system that should be delivered at the end of the project, they must have a common understanding of the definition of the concepts used. In contracts, definitions are mentioned, but as we will show, for instance, the word system is used in many different ways.

In this chapter, all concepts used in the first research question are discussed to answer the following sub-questions.

- How is a system defined?
- What is systems development?
- What is the definition of a project?
- Which artefacts can be distinguished?
- How is the end result described?

The word system itself is derived from ancient Greek: συ-στημά (Müller, 1969, p 707) and means a.o.:

- whole, composed of different members or elements
- system in mathematics, general sciences or music
- organisation, constitution, form of society.

The description of the various definitions of the concept “system” in the first paragraph is elaborate. This elaboration is intentionally, to point out that definitions in the different methodologies, contracting and other areas are not only unclear but also do not cover the same aspects concerning ISO standards.

When studying the definitions of systems in software development methods, it was notable that the definition changed over time: the newer methodologies as Scrum and RUP, do not use system definitions but mention "products" and "value-streams" instead of systems. These methodologies are described and compared in the second paragraph.

Projects and project development methodologies are described in the third paragraph since projects aim at changing a system. The fourth paragraph is about the artefacts of system development methods.

Two elements describe the end-result: quality and contracting.

In the last paragraph, the findings are discussed and reflected on. The conclusions of the discussion and reflection are used to build the first conceptual model.

3.2 Defining the concept ”system”

3.2.1 General Systems Theories

When describing IT systems, we noticed that the concept "system" is used in different ways and rather unprecise. Therefore, the general systems theories are consulted to obtain a good definition of systems to compare this definition with the definition of systems in software and other systems development theories.

An article of Mele, Pels and Polese (2014) provides a good overview of Systems Theories. They classify systems according to the question: “how does a system survive?”. This results in three schools:

- The Open Systems Theory,
- The Viable Systems Model,
- The Viable Systems Approach,

based on the General Systems Theory. In this General Systems Theory (GST) a system is seen as “a complex of interacting elements”. These elements behave differently when they interact with other elements than when they are single autonomous elements.

The “Open systems theory” (OST) considers relations between organisations and the setting in which they are involved. It focusses on the “organisations’ ability to adapt to changes in environmental conditions (with or without the need for information processing)”. Emery and Trist (1960) address organisations as socio-technical systems, underlining the two main components: a social component (people), and a technical component (technology and machines).

The Viable System Model (VSM), describes a system as an entity that is adaptable to survive in its changing environment. The viable system is an abstracted description that applies to autonomous organisations. This model was developed by the theorist Stafford Beer in his book Brain of the Firm (1972). A viable system is composed of five interacting subsystems which may be mapped onto aspects of the organisational structure. In broad terms, three systems are concerned with the 'here and now' of the organisation's operations performing functions, representing information channels to monitor and coordinate, structure and control. One system is concerned with the strategical responses to the effects of external (there and then), environmental and future demands on the organisation and one system is concerned with balancing the 'here and now' and the 'there and then' to give policy directives which maintain the organisation as a viable entity.

Finally, the Viable System Approach (VSA), suggests a new interpretation of consolidated strategic organisational and managerial models: Starting from this theoretical basis, the VSA has integrated several multidisciplinary contributions, applying them to the observation of complex entities. Principally, it has developed its theory around several key concepts derived by other disciplines: from system thinking (open system aspects), but also from information technology (specifically to IT roots based on cybernetics studies; Beer, 1975). VSA enables an analysis of the relationships that exist among an enterprise's internal components, as well as an analysis of the relationships between enterprises and other systemic entities in its environmental context. One of the fundamental concepts is that individuals, organisations, and social institutions are systems that consist of elements directed towards a specific goal.

Having a common goal is the realm of strategic thinking: In strategic thinking, a system is a set of processes that can be changed to improve the outcome.

Senge (1990) analyses how the systems method of thinking enables firms to become learning organisations. He looks at systems thinking, personal mastery, mental models, building a shared vision, and team learning as the basis for the development of three core learning capabilities: fostering aspiration, developing a reflective conversation, and understanding complexity to address value generation. From this approach, the organisation is seen as a holistic system, characterised by a high degree of integration between the factors intervening in the process of value creation.

Peters and Waterman (1972) indicate in their study “In Search for Excellence”, that the survival of companies is strongly connected with how the business goals are articulated (and how the system is viewed) : running a railroad-system or transporting travellers, for example: “Transferring people from A to B in a timely manner” instead of “running a railroad company”. Survival of the system will work better with the first goal: changes in (technical) circumstances do not influence the target. It does, however, require constant adaptation.

De Wit (2010) defines organisations more or less as a system: they distinguish the business level, the corporate level and the network level and the purpose of the organisation. (supra- and sub-systems as well as the common goal).

At a business level value propositions (products and services) are created through an activity system/value chain, based on resources/assets. How products are developed is based on the corporate level strategy, based on the positioning of the board of their organisation in an open system. In these days no system stands on its own, the network is essential as well.

In the Viable Systems theories, there is much attention for the information exchange between systems and layers in IT, according to the concept “Systems are open to connection with other systems for the exchange of resources. A system boundary is an adaptive element, containing all the activities and resources needed for the system's evolution.” The exchange between the Business-system and the IT-system is essential. In the next sub-paragraph, this exchange is described. In the last sub-paragraph, those viewpoints are combined as the start for the new model.

3.2.2 Business and IT

Henderson and Venkatraman wrote one of the most influential articles (Henderson and Venkatraman, 1993) on the strategic alignment between the business "system' and the IT system, which they call “domains’ In their article. They state that the IT strategy should be articulated in an internal and external domain. The IS domain contains at least three components: IS architecture, IS Processes and IS Skills.

A triangle from each corner shows a different perspective on the strategic alignment. In transformation projects, those four perspectives are combined: from the perspective of the competitive

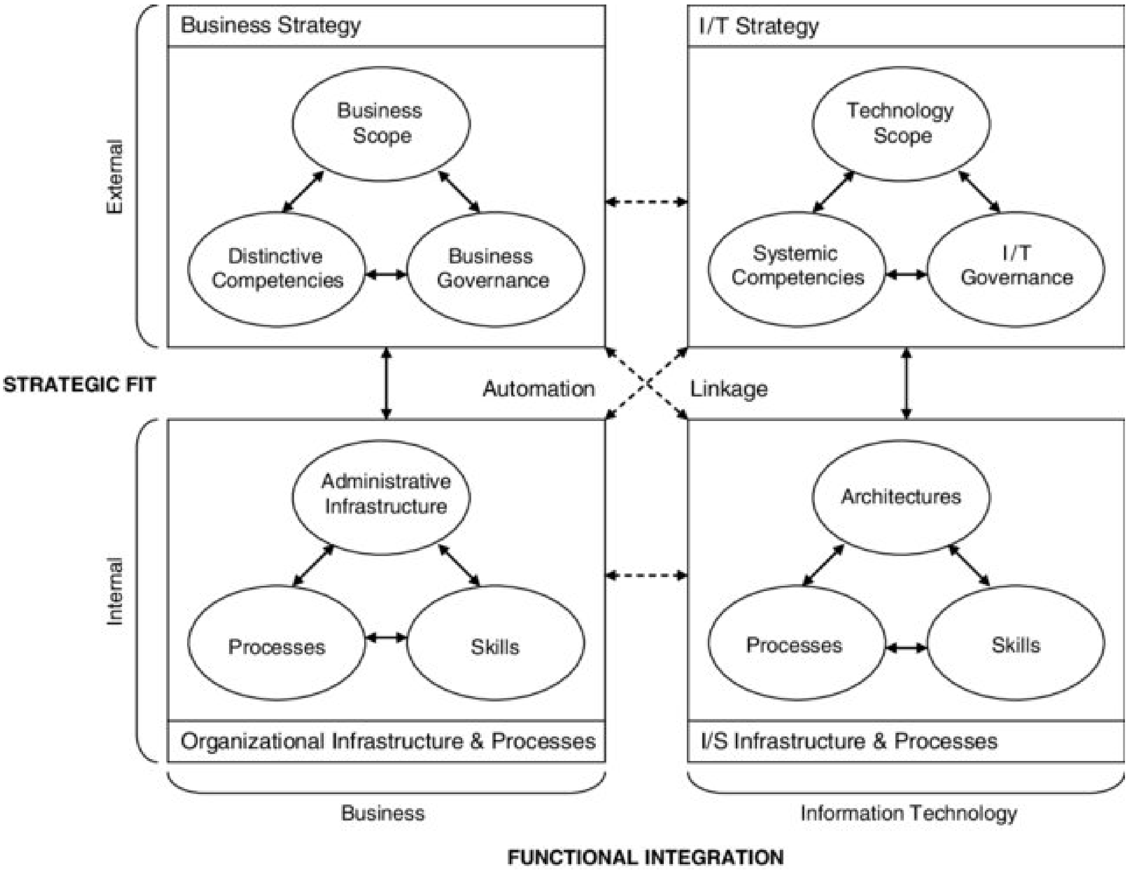


figure 2

potential alignment projects start, and it should meet the service level alignment when delivered. New technology is always at stake; thus, the perspective of technology is important but the alignment through strategy execution must be possible.

Rik Maes (2003) elaborated on this model in the “Amsterdamse raam werk voor informatiemanagement”, a nine-domain model in which the corners resemble the four areas of Henderson and Venkatraman .

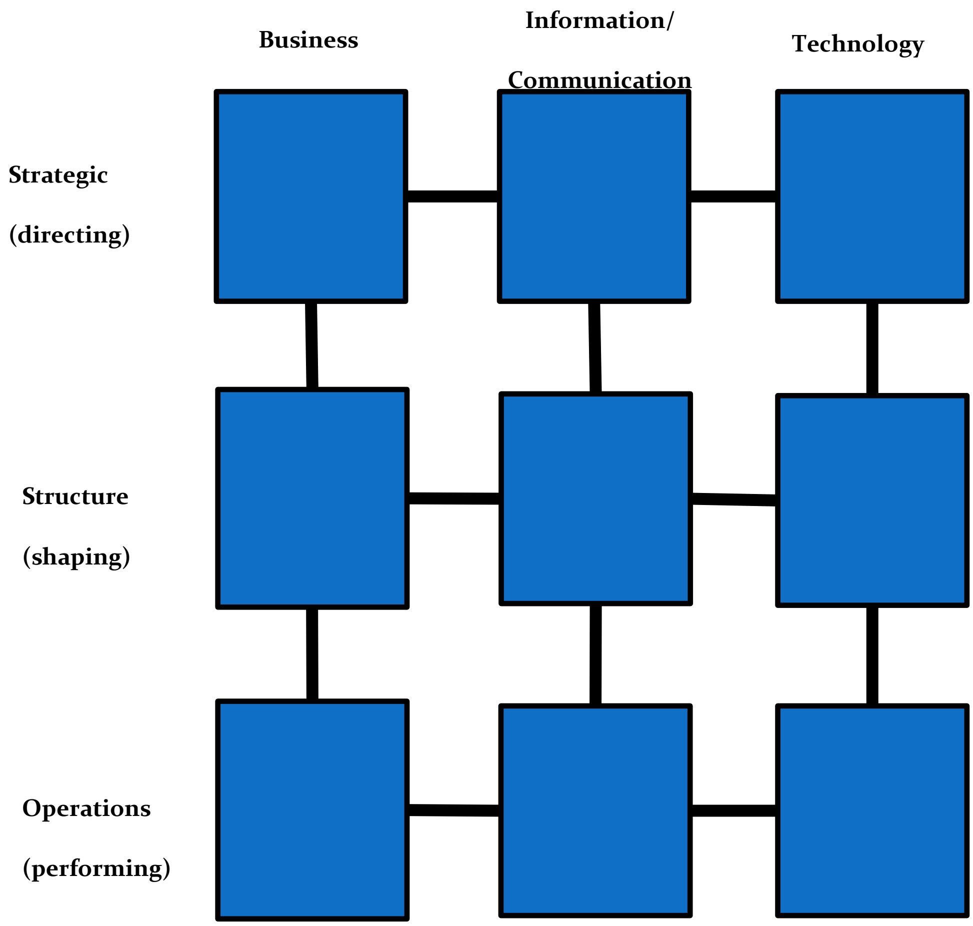


figure 3

It is explicitly not meant as an organisation model (Maes, website 23). The model tries to bridge the gap between business and IT also. The CIO-role is elaborated in the model. However, to deliver a project successfully, all nine areas must be covered.

The general system models and particularly the viable recognise "people” as a system as well. The IS models do not take this into account, only prescriptive as procedures or models. A considerable difference between product design and IT transformation is that in the latter the people mostly have to change as well to adapt to the new goals and the corresponding IT system.

3.2.3 IT and project methods

In almost 20 years, software engineering has changed rapidly. Cause and effects are intertwined: new software engineering tools and concepts improved the delivery time of software and made it possible to respond more quickly to the client's wishes. New agile methodologies have been introduced based on those new possibilities. However, the concepts of “rapid response technologies” introduce new challenges as for how to make the delivered software function in its context.

The definitions are described in 4 sub-divisions:

- The general standards as found in ISO12207 and ISO/IEC TS 24748
- The generic methodologies, as taught in universities
- The prescriptive methodologies (like Waterfall and RUP)
- The agile methodologies (to which Scrum belongs) defining products and value streams.

According to ISO standards

ISO12207 distinguishes four different processes to deliver a system: agreement, organisational project-enabling, technical management, and technical processes. The system concept is based on ISO/IEC TS 24748-1 where a system is described as a combination of interacting elements organised to achieve one or more stated purposes, with three notes:

- A system is sometimes considered as a product or as the services it provides.
- In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g. aircraft system. Alternatively, the word "system" is substituted only by a context-dependent synonym, e.g. aircraft, though this potentially obscures a system principles perspective.
- A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.

Generic methodologies

In the generic methodologies as described in Software Engineering handbooks, two classical (Aristoteles, Organon), different types of definitions can be found, definitions by describing the function or essentials together with the boundaries or differences with other entities or defining by summing up the constituent elements. The first way requires logical reasoning and is quite demanding. The second type can result in not summing up all the needed elements.

In "Software engineering", Sommerville (2001, p. 21) defines a system by describing a system as *"...a purposeful collection of interrelated components to achieve some objective*". The borders between systems he describes in terms of the system context: "the definition of a system context is not a value-free judgement. Social and organisational concerns may mean that the position of a system boundary may be determined by non-technical factors", thus using the first type of definition.

Pressman (2005, p 123-124)) uses the second type, first copying the definition of Webster's Dictionary:

- *a set or arrangement of things so related to form a unity or organic whole;*
- *a set of facts, principles, rules etc., classified and arranged in an orderly form so as to show a logical plan linking the various parts;*
- *a method or plan of classification or arrangement;*
- *an established way of doing something; method, procedure...*"

to conclude with his definition that a system is "*a computer-based system as a set or arrangement of elements that are organised to accomplish some predefined goal by processing information*". "The goal may be to support some business function or to develop a product that can be sold to generate business revenue".

Prescriptive methodologies

In Waterfall methodologies, the concept “system” is used as in Pressman, in the methodology Rational Unified Process, the founder Kruchten doesn’t give a definition, but by describing it, he actually uses the summing up way of defining: *“visualising, specifying constructing and documenting a software-intensive system demand that the system be viewed from a number of perspectives. Each of the stakeholders – end users, analysts, developers, system integrators, testers, technical writers, and project managers – bring a different agenda to a project, and each of them looks at that system in a different way at different times over the project’s life. A system’s architecture is perhaps the most important deliverable that can be used to manage these various viewpoints and thereby control the iterative and incremental development of a system throughout its lifecycle.”* (Kruchten, 2004, p 9-10).

However, Kruchten recognises a difference between a system and the software by making refences to “*a software-intensive system*”.

Agile

In Agile/Scrum, as Scrum is one of the leading frameworks, Sutherland does not describe systems but defines products that have to be made to deliver a "product" jointly. "*A product has to be implementable, catch on, and you must be able to become enthusiastic about it*". Moreover, "*you must be able to describe succinctly what you wanted to achieve when the product is finished*”. (Sutherland, 2014, pg 169-170). What exactly the product is, is not defined. It can mean an application, a training to use that application, a process design in which the application is used, trained people to use it or all of them. A product is meant to be used and has thus a purpose, but how the product contributes to the larger purpose is not described. Many agile methods, like SAFe (website 26), use value streams to drill down from purpose to products, which is essential in delivering business value through products. The way this contribution is guaranteed is through the process and the people in the process: deliverables are not prescribed. The idea behind it is that systems change continuously and that describing it, makes an organisation less adaptable to the future, thus emphasising the adaptation of a business, one of the systems in VSA. “*Value Streams represent the series of steps that an organization uses to build Solutions that provide a continuous flow of value to a customer. A SAFe portfolio consists of a set of development value streams, each of which builds and supports one or more solutions.*”. “*Each Value Stream produces one or more Solutions, which are products, services, or systems delivered to the Customer, whether internal or external to the Enterprise”.* In these definitions systems are part of Solutions which are part of a Value Stream. These definitions are based on the first typology of building definitions combined with the second since a solution can be a product, service or a system.

Project management methodologies

Project management methodologies aim at methodologically delivering a result. According to the systems theories, a project must deliver a changed system after completion. Therefore, it is interesting to study their definitions.

In general, three primary and open methodologies exist: Prince2, IPMA and PMi; the first two are mainly used in Europe, the latter in the US. Besides those open methods, several company proprietary methods are used. We do not discuss them here, because they are based upon the "big 3" and have specific add-ons for their own purposes.

PRINCE2 does not mention the concept of a system (Axelos, 2015). A business case, a project that should be aligned (Axelos, pg 234) and a plan that also consists of product descriptions that cover the products within the scope (Axelos pg 250) are mentioned to describe the change. They define the result by summing up the constituent elements. The definition can also be deduced from two themes: The Business Case and the Organization. The Theme Business Case mentions "Business Products" that have to be delivered "according to a business case" through a project. A system can deliver those Business Products. Also, scoping is a way of defining, limiting the object to be delivered. The Business Value afterwards has to be delivered in the organisation, which is a system in the Viable approach.

IPMA uses the ISO 9000 definition: A System is a set of interrelated of interacting elements. This definition does not set boundaries but describes a system by summing up elements like people, processes, products.

PMi uses the same ideas as Prince2: creating benefits and uses the word system for an information system, the project itself, products and parts of the information system.

Related methods

When taking into account the maintenance and management of systems the current methodologies BiSL/ASL/ITIL (best practice on the maintenance and adapting of business and IT systems) do not provide definitions of systems. The other framework, Cobit describes seven Enterprise enablers (Processes, Organisational structures, Culture, Ethics and behaviour, Information and Services, infrastructure and applications and People, skills and competencies). Both use the "summing up" way to describe what a system is.

For contracting, the Public Sector uses the Arbit (General Governmental conditions for IT purchasing) purchase conditions. Arbit defines a system as a product, meaning only the software according to a further described result: *“a Product is a good that the supplier delivers to the client according to the agreement. Software: the to be delivered lines of code by the supplier that can be used, directly or indirectly, in order to obtain a, further described, result. The software can be distinguished in standard or customized applications”.* Arbit (Arbit https://www.escrowalliance.nl/escrow-regelingen/arbit-escrowregeling/algemene-rijksvoorwaarden-bij-itovereenkomsten-arbit/) does not use the words “system” or “organisation”.

3.2.4 Reflections, conclusions and definitions

When comparing the VSA-approach of the General Systems Theories with the Amsterdam model, the VSA model is an interaction model, since systems can exist out of five different systems, whereas the Amsterdam model consists of the first three ("the here and now") systems, the lowest part. The system concerned with the strategical responses resembles the upper part and the system concerned with balancing the 'here and now' and the 'there and then' to give policy directives to maintain the organisation as a viable entity corresponds to the middle part. Both models are discussing the interaction and information exchange between the outside world, the business layer and the IT function.

The definition of systems in methods is somewhat ambiguous: systems, products, solutions, value streams are used to say something about what will be delivered by a project. In a picture, it is hard to draw the lines of a system. An enterprise is a system. A solution or product can be or be part of a system. A value stream itself is also a system and belongs to a system. Systems development and project management methods give different answers. The delivered value can only be measured by using the product, solution, system. If not used as intended, the value will not be delivered and last but not least: the assumptions on which the future value was calculated, could be wrong (Bradley, 2010).

This conclusion that these definitions could be unprecise corresponds with the finding of Melville (Melville, 2004). Their finding was that IT is valuable, but the extent and dimensions are dependent upon internal and external factors, including complementary organisational resources of the firm and its trading partners, as well as the competitive and macro environment. Melville states that *“linking IT to organisational performance is a complex problem that is informed by insights from "multiple theoretical paradigms"”*. In this research, we emphasise defining systems contributing to diminish this complexity. Senge brings in people who use the system as a factor: they have to understand and act according to the goal, by training, being enthused and participating. As a whole, they are a vital subsystem as well.

Reaching conclusions based on these findings:

- A common factor for systems, in IT as well as the general systems theories, is that systems can be distinguished by having a common goal
- VSA brings the insight that a system always is in connection with other systems
- VSA distinguishes levels in systems: supra and subsystems are added and useful: people are a vital system themselves
- A project delivers value to systems and subsystems by delivering systems, products and services that produce a change in the system(s)
- products, services and systems might have IT-components.

For this research we stipulate these definitions:

A system is a set of sub-systems in which interactions among enterprises’ internal components (sub-systems consisting of people, processes, tools) achieve a common goal in the outside world by connecting to other enterprises or organisations and influencing these systemic entities in their context (supra-systems) to survive.

We define an information system as one of the sub-systems:

An information system consists of processes, applications and technical infrastructure, meant to change, replace, support or steer (business) processes, by processing data into information for end-users, professionals and other systems.

An IT-system is defined as: *A system consists of applications and technical infrastructure to make this possible.*

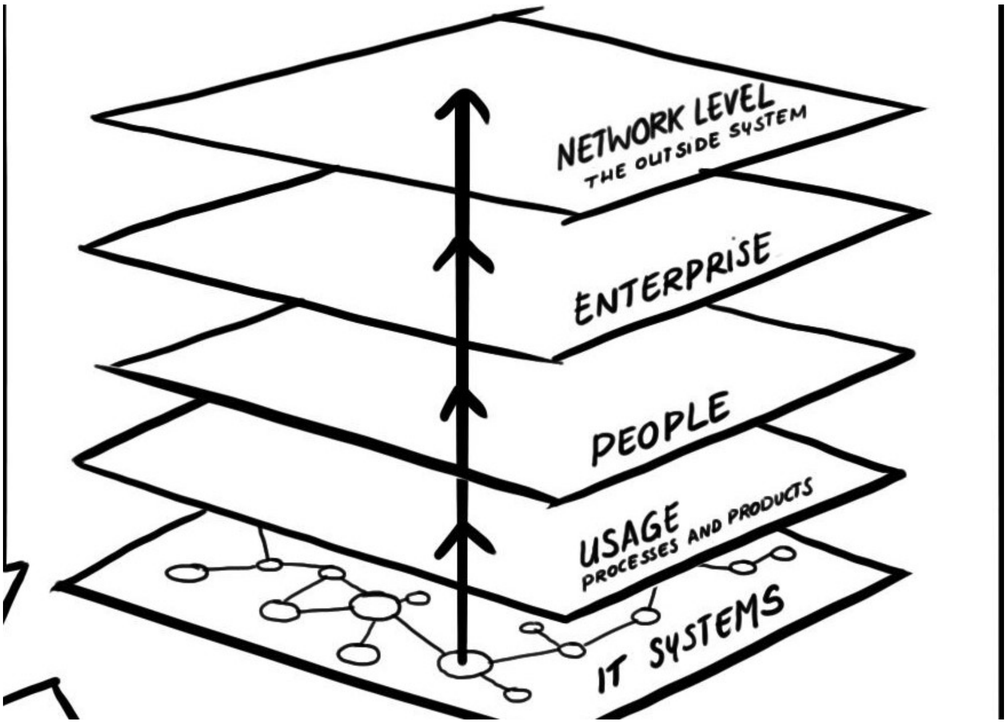


Figure 4

For this research we distinguish five layers in a system in which we combine the VSA for the people component, the B/IT alignment model and the Amsterdam model:

- Network, Context (the outside systems)
- Enterprise
- People
- Usage (Processes)
- IT Systems (Software and Infrastructure)

Contextual

In this layer, the quality of the interface design is essential, as well as the security and integrity of systems to which this system is connected and vice versa. When implemented the systems do not harm each other.

Enterprise

At this level, the planning and control of the system must be possible as well as the auditing of the system. The system must support all functional managerial tasks and manifests itself through products and services.

People

People use systems: managers, staff, users. Depending on their role the requirements are different: the functional suitability, performance, compatibility and usability are especially important. The new system should improve and support people after a learning period.

Usage/Processes

The processes that are designed to support the system in use are especially important when it comes to error handling, incident management and recoverability afterwards. These can be user-interfaces with the system or human procedures as well.

IT system

The IT system must fulfil the requirements (besides the ones that are described only by human-driven processes). It also has the task to adapt to the changing environment. The quality aspect "maintainability" is essential.

By these definitions and the layer-model the first sub-question “How is a system defined?” is answered.

3.3 System Development Methods

To create systems or change existing systems methods are used: when successfully implemented, all layers of a system change. Using a methodology implies a structured way to involve the stakeholders and the users of a new or existing system.

Since this research is about creating correct software for software intensive projects the software development methods (SDM) and project methods are described more elaborate than the methods belonging to the other layers. They are briefly touched because they belong to different disciplines and professions. For each layer, the current methods are mentioned to point out where they provide requirements the IT system should fulfil. The infrastructure methods are not described since no standard methods exist, and through new technologies (and building software correct) the infrastructure can be loosely coupled to the infrastructure.

Project management is the method to coordinate, plan and control the activities to implement the change from the requirements into a working system. The detail of the requirements determines the scope and is interacting with the method used. Therefore, we mention these methods as well.

In this section, the methods are described according to the five layers as pictured above.

3.3.1 Context

With “context”, the context, the place in the network of an organisation is meant as companies, public entities, where networks and alliances exist as well as competitors (De Wit, 2010, p. 365). Various causes can lead to a change in the surrounding system of the project: technology disruptions, new legislation, a new level playing field, changes in the society: The DESTEP-model (website 21 and Appendix 1.1) is a popular method to analyse those changes. In the EFQM model/Rhineland model the society is an essential factor as well and provides requirements as well.

The way organisations respond to these influences is related to its mission and vision (Peters and Waterman, 1972). The definition and purpose of the project should contribute to that mission and vision.

Marketing methods for companies and public affairs methods for public organisations are the methodologies to influence the external drivers.

The methodologies to influence the context (marketing, public affairs, influencing, lobbying) are not part of this research. For this purpose, we only consider the outcome of this external analysis in the way that they provide requirements that must lead to measurable results and effects in the outside world, such as:

- Laws as the foundation for public entities and other laws and rulings
- Mission and vision
- Interfaces to connect with systems in other entities
- Feedback from users of the system
- Customer demands

3.3.2 Enterprise

In this context, we define the enterprise level as the corporate level, the business level and the functional level as described in De Wit (2010). Corporate strategy is about how an enterprise internally is organised to respond at the network level.

From Mintzberg to De Wit, and many consulting firms have many different methodologies to change and describe organisations. It is impossible to discuss them since there is not a standard. For this purpose, we especially need this level to derive requirements and to prove that the requirements are met, when the system is functioning. Methods as benefits management, value management prove the system is delivering what it promised. This is followed up by controlling departments or planning and control departments (Bradley, 2016). The enterprise manifests itself by products and services.

For the IT system this level therefore delivers the requirements for:

- Functional suitability
- Performance efficiency
- Usability (operability, reliability, maturity, availability, fault tolerance, recoverability)
- Security (confidentiality integrity, non-repudiation, accountability, authenticity)
- Freedom from risk
- Strategic, tactical and operational information for planning, monitoring control and learning.

The following input is needed to derive requirements:

- The business functions
- The products and services belonging to these business functions
- The organisation chart
- The operating model
- The monitoring, control and learning model
- The budget/business case

Amongst others, the EFQM model provides a framework to describe these factors (Appendix 1.2).

3.3.3 People

People can transform the vision and mission into results, with the use of tools, as an Information System (processes and IT). A change caused by a project can create advantages and disadvantages thus creating fear or enthusiasm.

To implement a system not only the technical conditions must be met. People who have to use, operate, monitor and control it, have to grow accustomed to the new system.

The most used change models to achieve sustainable and measurable change (Robbins, 2011) are based on the conceptual models of Lewin's (the Three-step model) and Kotter (The Eight Step Plan) (see Appendix 1.3).

In software engineering methods, users are responsible for the validation of the system (in user acceptance tests). Users in an organisation are more than end users only. Strategic, tactical and operational management, system managers (functional, technical and operational), the service desk and other stakeholders have to ensure the system is functioning according to their needs. Each requirement needs criteria that can be measured and when the system unit is ready for use, have to be tested if it behaves as intended.

The requirements to ensure people can work with the system as is agreed upon at the enterprise level are (derived from ISO25010)

- Usability (Learnability, Operability, User error protection, User interface aesthetics and accessibility)
- Security (Confidentiality, Integrity, Non-repudiation)
- Satisfaction (Usefulness, Trust, Pleasure and Comfort)
- Freedom from risk (Health and Safety risk)

Design methodologies (as product design) are often used to obtain the requirements when Agile is used.

3.3.4 Usage

An important consideration in system development is whether the intended output with the available resources is met. A decision on the degree of automation versus the degree of human activity has to be made. The decision depends on specific requirements of the enterprise that are translated with the help of the ISO25010-framework to ensure a risk-free, satisfying and efficient way of working.

Processes contain these automated and human activities. They are derived from business functions and other models constructed at the enterprise level. The design of processes belongs to the realm of Business Process Management (Koorevaar, 2010), which covers Business Process Modelling (BPM), Business Process Re-engineering (BPR) and Business Performance Improvement (BPI). In the five-layer-model, these BPM methods serve the enterprise and the Usage layer.

From the many definitions, the definition of the Association of Business Process Management Professionals is chosen, relating these methods to the other methods needed to realise a whole system.

"*Business process management (BPM) is a disciplined approach to identify, design, execute, document, measure, monitor, and control both automated and non-automated business processes to achieve consistent, targeted results aligned with an organization's strategic goals. BPM involves the deliberate, collaborative and **increasingly technology-aided definition**, improvement, innovation, and management of end-to-end business processes that drive business results, create value, and enable an organization to meet its business objectives with more agility. BPM enables an enterprise to align its business processes to its business strategy, leading to effective overall company performance through improvements of specific work activities either within a specific department, across the enterprise, or between organizations*".

Common frameworks to guide these redesigns are the EFQM-model (also known as INK in the Netherlands) to improve processes continuously and LEAN, a methodology to improve operations.

3.3.5 IT methods

“Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software” (IEEE).

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high-quality software. SDLC is a process followed for a software project, within a software organisation. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. It is based on the ISO12207:2017 standard. Depending on the methodology the activities, if present, are grouped in the processes. Relating activities to specific roles, such as architects, business analysts, developers or testers is an organisational/governance issue, depending on the size of companies and their sourcing strategy. The e-CF Framework, provided by the European Union gives guidance on how these activities can be assigned to roles and functions. Professionalism is presumed to carry out the tasks well.

Software methodologies consist of prescriptive and agile methods. Each software methodology has its specific nomenclature and deliverables.

For this research three methods are examined:

- Prescriptive - Waterfall (as described by Pressman (2005))
- Prescriptive - Iterative (RUP)
- Agile (Scrum).

Waterfall is chosen because organisations change their way of working from Waterfall to Scrum/Agile as can be seen in one of the cases. In some cases, it is still a conventional method. RUP is selected because it is used in one of the cases and until 2014 widely used. Agile/Scrum is added because it is used in one of the case studies and is a predominant method at this moment.

Of course, lots of variations of this three methods exist, thus blurring the differences: Waterfall in various increments, RUP with increments, and Scrum in a SAFe (Scaled Agile Framework for Enterprises) resemble each other a lot. However, there are essential differences. The methods are described in more detail in Appendix 1.5.

Also, project methods to deliver the software are described.

3.3.5.1 *Software development methodologies*

ISO12207 divides software life cycle processes into four main process groups: agreement, organisational project-enabling, technical management, and technical processes.

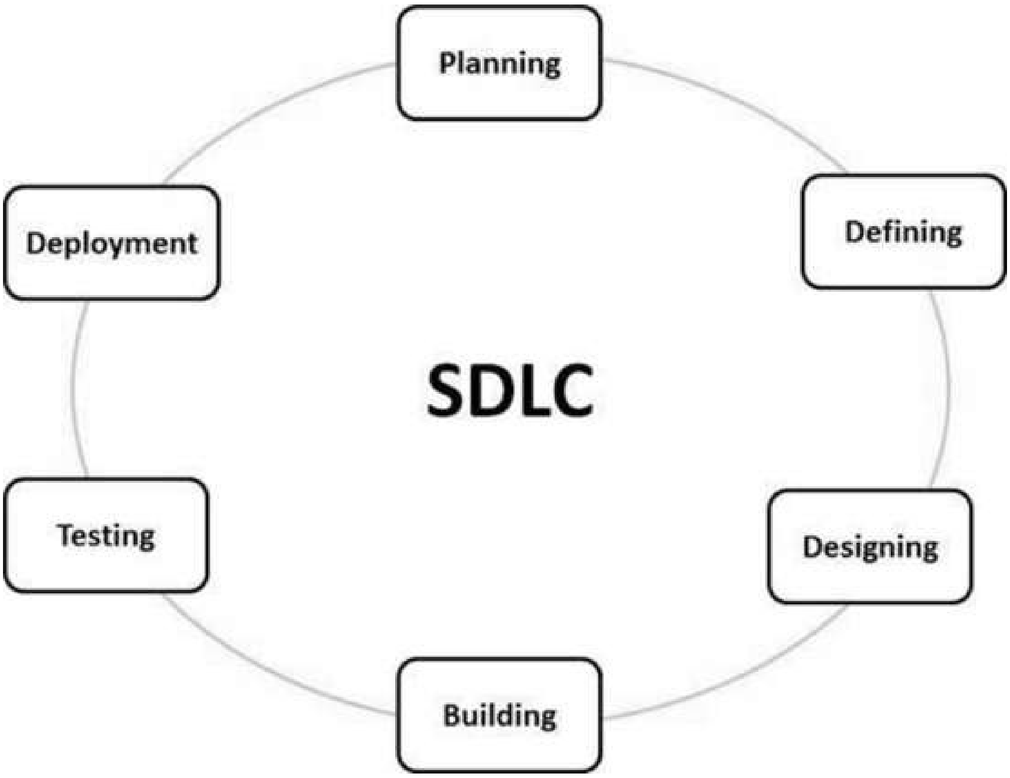


Figure 5

The software engineering process is described in the technical processes and contains eleven processes as shown in table 2 and 3.

The SDLC recognises six steps: Planning, defining, designing, building, testing, deployment.

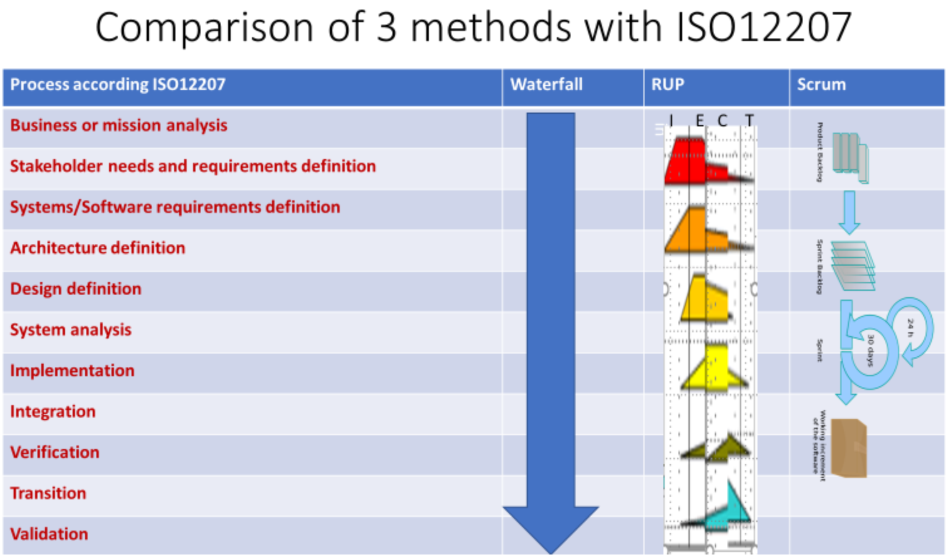
Table 2 describes the content of the activities. Table 3 shows the order in which the activities take place.

The schemes show that the prescriptive methods contain processes and deliverables (to be described in paragraph 3.4). Scrum does not prescribe deliverables. To deliver a full system all activities of ISO12207 need to be executed.

.

ISO12207	SDLC	Waterfall/Pressman/SDM	RUP	Agile / Scrum
Business or mission analysis	Defining	Definition study Requirements analysis	Vision document in Inception summarises the business requirements The Software Architecture Document is the result of the architecture definition on which the This results in the Software Development Plan.	The Product Owner provides the right user stories, prioritises (in the backlog for the Scrum team) and intermediates, refines or redesigns on behalf of the business owner
Stakeholder needs and requirements definition				
Systems/Software requirements definition		Global Design Requirements engineering Architectural Design		
Architecture definition				
Design definition	Designing	Design Engineering Functional Design	Use case models and elaborations in to define the SAD and the most riskful Usecases are realised Io prove the SAD. Elab, the others are elaborated in Construction	
System analysis		Technical Design, Component Level Design		
Implementation	Building	Build Realisation	Building of the software (in each phase possible)	Development & Deployment In sprints of approx. 2-3 weeks, delivers deployable software
Integration		Component Level Design Technical Design	In each phase the system is integrated further but mainly in Construction	
Verification	Testing	Testing Test	Prepared in Inception, Elaboration and partly executed in Construction and finished with Functional, Userand Production Acceptance Test in Transition	
Transition	Deployment	Deployment Transfer	Transition	
Validation	Planning →	In Use : repairing and checking results	//	OPS

Table 2



Comparison of 3 methods with ISO12207

Process according ISO12207	Waterfall	RUP	Scrum
Business or mission analysis			
Stakeholder needs and requirements definition			
Systems/Software requirements definition			
Architecture definition			
Design definition			
System analysis			
Implementation			
Integration			
Verification			
Transition			
Validation			

Table 3

In Waterfall, all activities are carried out in the order of ISO12207.

RUP presumes that the context, organisation, people and process layer are already defined and provides room to supplement the documents and gather them in the Vision document. RUP recognises 4 phases: Inception, Elaboration, Construction and Transition. Each phase may contain each activity of ISO12207.

Scrum focuses on building software: the team receives the specifications from the product owner. The product owner takes care of all previous activities. A framework of timeboxing (delivering work in a set time (2-3 weeks) is essential: it contains planning and defining (the backlog), build (designing and building) and testing(demo) and act (evaluation after deployment) phase.

3.3.6 Projects and project methods

The word “project” is derived from Latin: pro-iacere. It means to throw something forward. The word itself expresses that a result has to be accomplished in the future. Even if we speak of continuous improvement and small projects, projects are still present. All project management methods consider as constraints time, money, functionality and quality

In ISO/IEC/IEE 12207:2017 eight processes are distinguished as shown in Table 4 and compared to the “big 3” Prince2, IPMA and PMi. A more elaborate description of these methods can be found in Appendix 1.6.

ISO12207	Prince2	IPMA	PMi
Project Planning	Starting up a project Initiating a project Controlling a stage Closing a project	Plan Do Check Act	Initiating Planning Executing Monitoring and Control Closing
Project Assessment and Control	Controlling a Stage	Check	Monitoring and Control
Decision management	Manage by exception Roles and Responsibilities	Do	Execution
Risk Management	Risk	Check/Act	Monitoring and Control
Configuration Management	Change	Do	Execution
Information Management	Roles and responsibilities	Do	Execution
Measurement	Progress	Check/Act	Monitoring and Control
Quality Assurance	Quality	Check/Act	Monitoring and Control

Table 4

Each methodology defines a project as “*a temporary organisation to reach a certain goal that is described in a business case*”.

All methodologies have a similar way of controlling a project: all have an initiating, executing and closing phase. The description of the governance of a project is different. In the initiating phase, the requirements are supplemented depending on the level of the contract.
All phases provide certain artefacts that are input for the Project Plan. This Plan is the guiding

document. In this Project Plan, all activities within the scope are described to meet the project goals and to ensure that the benefits of the project can be measured afterwards.

3.3.7 Reflections and conclusions

A system contains many levels of which five are elaborated for this study. From all levels, requirements originate that need to be met to create a working system. These requirements are a product of the change plans in those levels. To achieve business value by a successful implementation at all levels, not only the activities to obtain the requirements must be coordinated, the activities to change the levels have to be ready at the due date as well. Then the IT system can function adequately in its environment.

In the SDLC this is called the Plan and Define phase. Depending on the methodology the project starts further in the cycle. The question is, if this Plan and Define (and for Scrum also Design) activities are part of the project or not. If a project board wants to start a project, the question is what they ask for: deliver a change or deliver according to the requirements that were prepared in advance.

Summarising:

- Some methodologies provide more structure to intertwine with the other system development methodologies than others. For instance, Scrum expects the Product Owner to know how the requirements are gathered and efficiently presented to the team.
- All project methods have room for initiating a project and gathering required input.
- Depending on the methodology and the contract, the client has to prepare more or fewer artefacts before the delivery phase in the project can start.
- RUP and Scrum (other methods have not been taken into account) do not cover the same activities of ISO12207.

Primarily how to obtain the business requirements, analyse them and provide an architecture is less present in RUP and not present in Scrum.

Project management is not only needed to deliver all aspects for use at the same time. Some form of project management is needed in the initiation phase of the project to bridge the differences of the methods with ISO12207 to start the design and build with the same quality of specifications, The initiating phases of the project management methods have no intention to perform a full definition phase. This add-on seems an essential element.

3.4 Artefacts and End Result

Artefacts are the tangible results of a project, but they do not represent the effect of a project. Not the comprehensive documentation, nor their presence itself is essential; instead, the quality of these artefacts is. The real result, i.e. the effect of the change, is measured afterwards when the outcome of the project is used.

The presence and the quality of these deliverables are dependent on the agreements in the contract.

The quality of the end result of a project can be described as: delivered as intended and delivered as contracted.

The quality of the effect of the project will not be discussed since this is a different research topic.

In this section, the following themes are described:

- Artefacts of a project
- The quality aspects of software
- Contracting

For this research, these aspects are attributed to the f*ive system layers.*

The perceived quality of the fulfilment of the contract is not only depending on the quality of the specifications in the contract but also on the quality of the software itself including the documentation of the system.

ISO25010 is a standard to deliver an intrinsically correct IT system. This standard implies that a client does not have to ask for it; the supplier should deliver software according to these standards unless otherwise contracted. On the other hand, this implies that the Client has to deliver the correct content of the requirements. ISO2510 contains eight quality terms for the product itself and five items in using the product (as detailed and explained in Appendix 2).

ISO12207 describes the necessary activities for software engineering. For project management, we use the same standard. The ISO standard 15288 contains the SDLC too. These standards are recently calibrated (Appendix 1.5).

For artefacts, no standard is available except the standards on documenting itself (ISO/IEC/IEEE 26531:2015). The artefacts depend mainly on the choice of the SDM, size of the project and the quality system of the organisation that uses the documentation.

3.4.1 Artefacts

This research started with the question whether some deliverables are more important than others for project boards.

In table 5 on the next pages, the artefacts of the three methods are compared with the five levels that are used in this research to define a system.

In each area is described which documents are available in which method. In each layer activities of software engineering are repeated, the subject is different, the activity is the same.

Level	IS012207		METHOD	Kolom1	Input from/used by
		Waterfall	RUP	Scrum	
Contextual	Business and Mission Analysis	Definition study Business Object Model Architecture Vision	Vision Document	Product owner	Marketing and/or Public Affairs/Legal Board of directors
	Stakeholders needs Requirements definition	Requirements model	Vision Document	Product owner	Marketing and/or Public Affairs/Legal Finance & Control
	Systems/Software requirements	Requirements model	Vision Document	Product owner	All stakeholders for this level
	Architecture Definition	Architecture Requirements	Software Architecture Document (SAD)	Product owner	All stakeholders for this level
Enterprise	Input upper level	Business Use Case Model Business Architecture	Business Use Case Model	Product owner	Corporate Strategy Product management (product and services) Business Process Improvement
	Design Definition	Use Case Model	Use Case Model	Product owner	Planning & Control, KPI's Business Process Redesign
	Design Definition	Information Model	Vision Document	Product owner	Quality, Security, Privacy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	Process Management
People	Stakeholders needs Requirements definition	Use Case Model	Use Case Model	Product owner	Business Process Redesign HRM Training Plans, UXP, Inclusion, job satisfaction freedom from risk
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	
Usage Processes	Stakeholders needs Requirements definition	Use Case Model Interaction Models	Use Case Model Supplementary specifications Interaction Models	User stories in Product Backlog	Business Process Modelling Control/Risk: security/error protection
	Design Definition	Use Case Model Interaction Models	Use Case Model Interaction Models	User stories in Product Backlog	HRM: job descriptions and training plans Security: authorisations Control: compliancy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	

Level	IS012207		METHOD	Kolom1	Input from/used by
		Waterfall	RUP	Scrum	
IT	Systems/Software definition	Use Case Model Interaction Models	Use Case Model	Backlog	Operating Model, Governance, Security and Maintenance
	Architecture Definition	Information and Technology architecture	Software Architecture Document (SAD)	User stories in Sprint Backlog	Operating Model, Governance, Security and Maintenance
	Design Definition	Use case specifications	Use case specifications + Additional requirements specifications	User stories in Sprint Backlog	Sprints or build teams and maintenance
	Systems Analysis	Interaction Design and DB-design	Use case realisation, Class diagrams, Activity Diagrams. State transition diagrams	User stories in Sprint Backlog	for the above and Operations
	implementation Integration	Build, component or package	Build Implementation model	Working software	for the above and Operations
	Verification	Systems test	Test Implementation Test Results	Working software	for the above and Operations
	Transition	Production Acceptance Test	Test Implementation Test Results	Working software	for the above and Operations
	Validation	User and Business Acceptance	Test Implementation Test Results	DoD	for the above and Operations

Table 5

3.4.2 Quality of software/systems

Typology		*attribute*	IT system	Usage	People	Enterprise	Context
Product Quality							
	Functional suitability						
		Functional completeness	RI	RI	RU	R	R
		Functional correctness	I	I	U	R	R
		Functional appropriateness	I	I	R	R	R
	Performance efficiency						
		Time-behaviour	I	I	U	R	R
		Resource utilization	I	I	U	R	
		Capacity	UI	I		R	
	Compatibility						
		Co-existence	I	I		R	R
		Interoperability	I	I		R	R
	Usability						
		Appropriateness recognisability	I	I	RU		
		Learnability	I	I	U	R	
		Operability	RUI	I	RU		
		User error protection	I	I	U	R	
		User interface aesthetics	I	I	RU	R	
		Accessibility	I	I	RU	R	
		Reliability	I	I	U	R	
		Maturity	I	I	U	R	
		Availability	I	I	U	R	
		Fault tolerance	I	I	U	R	
		Recoverability	I	I	U	R	
	Security						
		Confidentiality	I	I	U	R	U
		Integrity	I	I	U	R	U
		Non-repudiation	I	I		RU	U
		Accountability	I	I		RU	
		Authenticity	I	I	U	RU	U
	Maintainability						
		Modularity	RUI				
		Reusability	RUI				
		Analysability	RUI				
		Modifiability	RUI				
		Testability	RUI	IF	U	RU	U
	Portability						
		Adaptability	RUI				
		Installability	RUI				
		Replaceability	RUI				
Typology		*attribute*	IT system	Usage	People	Enterprise	Context

Quality in use							
	Effectiveness			I	U	R	
	Efficiency		I	I	U	R	
	Satisfaction						
		Usefulness	I	I	RU	R	
		Trust	I	I	RU	R	
		Pleasure	I	I	RU	R	
		Comfort	I	I	RU	R	
	Freedom from risk						
		Health and safety risk mitigation	I	I	RU	R	
		Environmental risk mitigation	I	I		R	R
	Context coverage						
		Context completeness	I	I	U	R	
		Flexibility	I	I	U		R

Table 6

Table 6 contains the quality aspects of software as mentioned in ISO25010. For each specific quality attribute is clarified if it provides requirements (R), is used (U) or implemented (I) and in which layers.

3.4.3 Contracting the end-result, agreement processes

Contracting with a Supplier, whether internal or external, is mentioned in the ISO/IEC/IEEE 12207:2017 standard, which includes the procurement and supply processes, which contain the activities related to establishing an agreement between a supplier and acquirer. The acquisition process corresponds with all the activities involved in initiating a project and consists of different activities and deliverables.

Contracting of IT-companies seems complicated as Rooks (2002) proved: almost 75% of the contractors perceived the proposal as too good to be true. In real life, over 50% of the projects did not deliver on time and within budget (on average 89% exceedance). It underpins the Standish Report (2015)

In the construction industry exist a limited number of contract types that mitigate these risks (van Vossen, 2014), and in IT-contracting many of these have been followed:

- Design, Build, Operate and Finance (Shared revenue)
- Design, Build, Operate (Business Process Outsourcing))
- Design, Build, Turnkey delivery (Ready for use)
- Build by Design (fixed price)
- Appointing a project manager and hiring expertise
- Hiring people with the needed competencies at the lowest cost

For IT projects (borrowed from Capgemini) five types are identified since the first and second contract form have in common that the Supplier is responsible for the outcome of the project

- Fit for Purpose: realising the benefits case (to the company and supplier's company),
- Fit for use - also turnkey – (people, process, IT)
- Conform specs at fixed price or date – realising correct software according to requirements
- Hiring Expertise – Knowledge of specific areas
- Hiring Capacity -- hiring people with skills that are obsolete in the company

The influence and responsibility of the Supplier on the outcome of the project decreases in the order of the list above. For the Client, it is precisely the other way around.

These types of contracts correspond to the five system layers:

Contract	Covers	Responsibility	
		Supplier	Client
Purpose	all system layers	To run the business	Provide a well-defined result
Use	People, Usage, IT	Deliver working processes and trained people	+ Provide Business model, Checks and Balances
Specs	only IT	Deliver according to specifications	+ Provide specifications
Expertise	Solutions in that area	Deliver a solution	+ Deliver a working architecture
Capacity	Professionalism	Deliver capacity	+ Deliver work packages

Table 7

According to the Brinkman Arrest, a Supplier is responsible at all levels, if the Supplier accepts the requirements of the Client without assessing if the result can be met.

When hiring a project or program manager the same contracts can be applicable regarding responsibility.

3.4.4 Reflections and conclusions

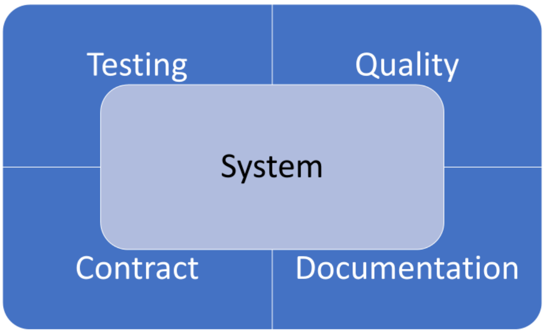


Figure 6

To ensure that the requirements of the contract are met, it is necessary to ensure by testing that the software has the right quality. To prove the correctness by testing, documenting the requirements is necessary.

When the system is in use documentation is needed to reduce the cost of maintenance and to understand the operation of the system (by the end users and operations).

The artefacts of projects somewhat differ per method:

- Agile mentions in the Agile manifesto "Working software over comprehensive documentation”. What and how the product owner is delivering cannot be found. There is no definition of artefacts to be delivered and methods to be used except working software.
- RUP assumes that the outcome of the contextual and organisational layer analysis is present and can be included in the Vision Document.
- Waterfall (Pressman) provides for each layer artefacts.

The requirements are derived from all layers, and it has to be ensured by the project leader to the project board that all requirements are considered and how.

Herewith we give a provisional answer on the question “How is the end result described?”

3.5 Findings, reflection and conclusions

In this chapter, the following questions are discussed

- How is a system defined?
- What is systems development?
- What is the definition of a project?
- Which types of artefacts can be distinguished?
- How is the end result described?

Each part of the definition of the constituent elements of the problem statement led to findings and conclusions. The most important findings and conclusions from these sections are:

1. The concept of "system" in software development, project and maintenance methodologies is defined differently or not defined at all.
2. Systems development is more than IT-development alone.
3. The definition of systems in supra- and subsystems gives a good insight into how software development methodologies cover the layers of the system
4. The layers of a system are comparable with levels of contracting.
5. Sustainable change can only be reached when all levels are changed with the same purpose.

Answering the first sub-question, "how is a system defined?", we use a threefold definition:

> *A system is a set of sub-systems in which interactions among enterprises' internal components (sub-systems consisting of people, processes, tools) achieve a common goal in the outside world by connecting to other enterprises or organisations and influencing these systemic entities in their context (supra-systems) to survive.*

An information system as one of the sub-systems is defined as:

> *An information system consists of processes, applications and technical infrastructure, meant to change, replace, support or steer (business) processes, by processing data into information for end-users, professionals and other systems*.

Also, an IT-system is defined as:

> *A system consisting of applications, maintenance and quality processes and technical infrastructure to make the above possible.*

The question "What is the definition of a project?", is answered as follows

> *A temporary organisation to reach a certain goal that is described in a business case*".

To reach this goal, understanding that a system tends to survive, means that organisations who want to establish a sustainable change have to address (and change) all layers of the system with the use of all methodologies. The question is: which part is included in the project and which part will be delivered by the organisation itself.

Methodologies are not covering the same layers and aspects. Methodologies are covering different parts of the ISO12207 activities.

In project management, it means that the initiating phase has to cover different activities, depending on the methodology. All methods are compared to the ISO12207 norms

Software methodologies do not provide methods for the change of people, processes, organisations or the environment in which an organisation operates. Some software methodologies give more guidance than others to relate to those other aspects.

Some documentation is needed, to prove the correctness of the delivered software. Testing is necessary to prove that the contract is fulfilled by meeting the requirements. Methodologies have different documentation standards. Scrum has no specific guidelines for it, besides the Agile Manifesto sentence, "a working system over comprehensive software" and the process of Demos. The Product Owner has to take care of that. A contract describes at what level the system will be delivered. The responsibility of the Client and Supplier are communicating vessels regarding the level of the contract.

If it is necessary for project boards to obtain knowledge of these incongruencies or that the model provides enough insight for them, will be answered by the questions that are left over from the sub-questions:

- Which artefacts seem crucial?
- What do project boards/clients steer/what question should/could the client ask? And
- Can a model/questionnaire be derived for the client/project board?

The effect of a project can only be measured afterwards, by benefits realisation management (Bradley, 2016) but one essential part of achieving these benefits is that the project will deliver the proposed artefacts according to the forecasted budget, at the proposed time. The other essential part is that the effect the project should have can be assured and assessed throughout the software development cycle.

In the next paragraph, we will start with the first derivation of the project board guide/model.

According to the design science model, all foundations now have been laid serving as the starting point for the construction of the model.

4 A first Iteration: Develop, Build, Justify and Evaluate.

4.1 Introduction

In the previous chapter, according to the design science methodology, the knowledge base is developed. In the introduction, we discussed the relevance of this study. Many B/IT-projects are failing. However, the evaluations of those projects mainly concern the quality of process and reporting (Bronsgeest,2016), not the quality of the products and the steering upon that quality.

In this chapter a first conceptual model is constructed, evaluated and tested to start answering the sub-question "Can a model/questionnaire be derived for the client/project board?".
The question "What do project boards/clients steer/what question should/could the client ask?" will be answered entirely after the model is elaborated and validated in the next chapter. This chapter contains the preliminary answers on that question.

The underlying hypotheses of this research are:

- Projects are not delivering results because clients in steering committees lack knowledge of the SDLC (IT).
- The assumption is that some artefacts are more important than others and that
- The choice of methodology plays a role in this.
- ISO25010 is no input for the requirements.

The justification for the first three hypotheses is based on the literature study. The outcomes of the literature study to develop the model are:

- Systems have many layers (we chose five layers).
- Developing a system that provides the intended result requires a holistic approach thus developing all systems aspects at all layers.
- Some methodologies provide more structure to intertwine with the other system development methodologies than others.
- All project methods have room for initiating a project and gathering required input.
- Contracting levels define the distribution of work between Client and Supplier.
- Depending on the methodology and the contract, the client has to prepare more or fewer artefacts before the project can start.

In paragraph 4.2 The first conceptual model can be found.

This model is evaluated by comparing two case-studies with the model (in section 4.3). These two case studies were selected because:

- Both were debated in the public domain. Materials of these projects were well available.
- The researcher was not involved in these projects, nor clients or suppliers of the researcher.
- The projects are different in many aspects:
    - Contracting: one concerns a private client and a private software company; the other contract is between two public entities.
    - System Layers: in the public project all layers are essential; in the commercial, they seem less critical since it is about building a solution.
    - Methodology: the commercial project has been performed by a prescriptive methodology (RUP) and the second by an Agile (Scrum).

Both projects have been made anonymous for confidentiality reasons. The project in the Public Domain is called Public, the one in the private domain Private, the client is named Client and the software provider Supplier. The underpinning reports and deliverables are available for inspection.

Each case is described according to the initial model, and the initial model is evaluated after each case.

Paragraph 4.4 contains the result of the survey by which all hypotheses are tested, especially the first. Some questions are added to justify the model.

The last paragraph, 4.5, evaluates the model by reflecting upon the findings of the cases and the survey. It contains proposals to adjust the model.

4.2 Develop and build: an initial conceptual model

The initial model is a simple model to analyse the two case studies and to find out which viewpoint is the determining factor. In the previous paragraphs, the following aspects have been found as constituent elements to deliver a successful project or change:

- System level
- Contract level
- Methodologies for software engineering
- Methodologies for project management
- Quality aspects
- Artefacts
- Benefits realisation (Time and Money)

These aspects lead to the following possible elements for the model:

System level

The system level is an independent element: it cannot be derived from another element and therefore is required in the model. People, processes, people and the organisational structure also have to be considered. The three system parameters, variety, variability, and indeterminacy (Barile and Golinelli, 2000) in all these areas are essential. These parameters are covered by the full systems development methodologies (IT, processes, people, organisation, context).

Contract level

The initiation phase of project management is meant to gather all requirements according to the level of contracting. Technically speaking no contract can be concluded if the requirements needed to execute the contract are not present yet. Contracting and project management are dependent of each other. The project always has to start at the right requirements level. Contracting is thus an essential element.

The contracting levels are also subdivided in five levels (capacity, expertise, conform specs, ready for use and delivering business value). Apart from the capacity and expertise layer by which projects are realised, these layers resemble the management layers (people): operational for the detailed specifications, tactical for the right usage and process improvement, and strategic for the goal-setting and the decisions on the migration from the as-is to the to-be.

Methodology

Since SDM's do not provide the same input for the execution of the contract, this must be taken into account as another critical element. Project methodologies have similar phases and are thus less decisive.

Outcome

The outcome of a project is measured by the degree of realisation of the benefits when in use. This measurement also requires the quality aspects concerning the needed time and money to realise these benefits (outcome minus costs). The considerations to deliver at a certain quality in a specific time and for a specific amount of money is documented in the artefacts and results in the contract. The outcome is the result of the project and is therefore not an element in the model.

The model

The conclusion is that the model to evaluate and assess the two case studies consists of the three elements system level, contract and method.

For this moment they are used in this order since it is not known yet which is the prevailing one.

Since this model has to guide business owners/project boards to ensure the quality of their projects, the initial model aims at finding the starting point for assessing the situation when a project board is steering a project.

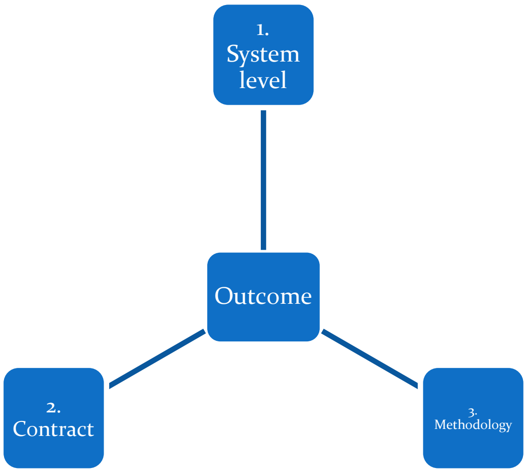


Figure 7

From the review of the Elias-committee, it is known that the knowledge of the client/business owner is not always sufficient to judge the situation as Elias (2014) describes thoroughly. It would be no problem if the client and supplier have an open and transparent relationship and inform each other. In reality (as among others Elias (2014) and Rooks (2010) show us), this is often not the case.

The case studies have the following structure:

- the system-level and its artefacts
- the contracting level
- the system development methodologies.

4.3 Justify by case studies

4.3.1 The Private project

The start Private started as a project after a huge breakthrough for Client. A leading consulting firm promoted Clients' application to a reputable organisation in their sector as an up-and-coming technology solution in their area. This organisation bought its application under one crucial condition: the software should be refactored to a state-of-the-art application. This condition implied rebuilding the package in the newest Microsoft c#-version. This refactoring became more interesting since other organisations also got interested and signed contracts, under the same condition, that the software should be state-of-the-art. Clients' Business Case was built on the signed contracts, and the assumption other contracts would follow. This was a giant leap: the current turnover was 40-80 times lower.

The Business Case was supported by signed contracts of well-known companies in that sector. Essential for the Business Case was obtaining a first minimum viable product to generate revenue for future developments. Contracts with customers were signed; one of them was executing the acceptance-test.

Several software companies were asked to propose a solution for migrating the application to another platform. One of them convinced Client that the software should not be migrated but had to be rebuilt totally because the requirements had been changed by offering the application to other branches in their sector. This advice implied building a new kernel to support the different branches, and add functions per branch, which could be sold separately to create a better basis for future additions. The data handled by this application is specific, personal and precious. Journalists and companies in the branch would do a lot to obtain those data.

Contracting the work The chosen contract was a ”capacity” contract and signed in November 2005. The reason behind this choice was that Client obtained a bad risk rating. Such a contract offered the possibility to stop at any time. However, the contract was closed for three years.

Supplier proposed RUP as the methodology to use, delivered a project manager and a team. The management of Client just had to accept the products and review them timely (according to the Vision document (Appendix 4.1.2)). The addendum of the contract, however, shows a different approach (SDLC in general, with distinct phases, starting with phase zero) and not RUP alike. The wordings of the documents, however (Vision Document, SAD) are RUP-terminology. The Phase 0, mentioned in Appendix 4.1.1, pg 15 and further), says that necessary documents would be completed (in approximately 8-12 weeks), according to "People, Process, Technology" resulting in a Delivery Handbook. Depending on the outcome of this Phase 0 the contract could be amended.

In November 2006, a year after the start and three months after the first due date (not met) of delivering the software to the acceptance team, Supplier pointed out the stipulations in the contract (exhibit 16) extra clear: Client had to steer the project on budget, planning and deadlines. Supplier only had a best effort obligation. A few months before, Client had behaved accordingly but was summoned not to do so. Supplier still delivered an Engagement Manager.

Execution Suppliers' project manager and the software architect created the SAD, Vision Document and Software Development Plan. This Vision Document and SAD and Software Development Plan (Appendix 4.1.4) were input for the contract, according to RUP. The SDPs' last version is 0.75, and the SADs' is 0.9. In this case, the project manager of Supplier wrote all documents and gathered the requirements (product 3 in Progress Reports). The approval of the documents was slow (and often referred to in the progress reports). However, the construction of software in India had already started. The factory was set up in January and already produced software. The first increment has to be delivered in August 2006 to start acceptance testing.

According to the contract, Client was responsible for delivering the Business Use Cases and the requirements, but he hired capacity from Supplier to describe the Use Cases based on the requirements of Client. Client was slow in validating and was summoned to review faster. Suggested expert professionals mentioned in the contract are only technical. No change or business consultants are involved. Client did not recognise his requirements in the delivered documents but did not deliver input themselves either. Project reports indicate that complaints were made about lacking requirements, but the fulfilment of the contracts still was reported as okay. The project went on, even programming started already, without a good definition of the business model. Instead of requirements one of the two applications was delivered from which the requirements must be derived.

It is unclear in which Phase the project was. After the Elaboration, a go-no-go decision could also have been taken. In the reports, much was said about the problems occurring, but the smileys indicate a well-run project.

According to RUP (Kruchten, 2004, p168-169), the project should have been paused or stopped at the end of the Inception Phase. The go-no-go decision was clear: all necessary documents were lacking. However, the construction of software already started in India after the first PoC (Proof of Concept) was made in the Netherlands (as shown in Appendix 4.5, “productïe” 4). At least at the 14th of March a team was already producing.

Client signed a contract with Supplier at the end of 2005 for three years. The project ended in 2009 after the absence of payments. Client states that no shippable software was delivered until then, confirmed by a customer of Client who was acting as the primary user who conducted the User Acceptance Test (UAT).

4.3.1.1 *Contextual Level*

Artefacts derived from the contextual level

Level	IS012207	METHOD			Input from/used by
		Waterfall	**RUP**	**Scrum**	
Contextual	Business and Mission Analysis	Definition study Business Object Model Architecture Vision	Vision Document Business Case	Product owner	Marketing and/or Public Affairs/Legal Board of directors
	Stakeholders needs Requirements definition	Requirements model	Vision Document, Business Use Case Model	Product owner	Marketing and/or Public Affairs/Legal Finance & Control
	Systems/Software requirements	Requirements model	Vision Document	Product owner	All stakeholders for this level
	Architecture Definition	Architecture Requirements	Software Architecture Document (SAD)	Product owner	All stakeholders for this level

No reference is found in any document to such documents as business planning, marketing planning the Business Case or other products that give input for the contextual level.

The project manager of Client wrote in the Vision Document:

"The Vision Document provides a complete vision of the software system under development and supports the contract between the funding authority and the development organisation".

In this case, a Vision Document (Appendix 4.1.2) was present. It was signed off at the 13th of March 2006, whereas the contract was signed at the 3d of October 2005. Essential for the Business Case was obtaining a first minimum viable product (MVP) to generate revenue for future developments by pleasing the first customers. The MVP should be delivered in August 2006. One of the buyers was responsible for the acceptance test. The buyers were not mentioned as stakeholders, only as users. Not reaching the MVP in August was a real risk. It was not addressed. A risk list is not found except the one in the SDP.

No contextual Enterprise Architecture can be found, describing the interactions between organisations (Clients' and buyers of the solution and others).

The requirements gathered from this level are not present.

Methodology The only methodology used for this level is RUP, which is, however, not equipped for the contextual level. Methodologies for Business Architecture, business consultancy or marketing, should lead to instances of the Vision Document, the Business Uses Cases and the contextual Architecture which had to be delivered or approved by Client. These processes are not part of RUP, but the outcome is supposed to be present to start the business analysis.

Contracting At this level, a business consultant or business architect could have helped to clarify the business context and issues. They were not hired nor suggested to be involved; In the contract itself, only technical capacity and project management were mentioned. This level requires other competencies as well.

Findings: At this level, no input was provided to create the outer structure of the solution. Other competencies were necessary but not provided.

4.3.1.2 *Enterprise level*

Artefacts derived from the enterprise level

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
Enterprise	Input upper level	Business Use Case Model Business Architecture	Business Use Case Model	Product owner	Corporate Strategy Product management (product and services) Business Process Improvement
	Design Definition	Use Case Model	Use Case Model	Product owner	Planning & Control, KPI's Business Process Redesign
	Design Definition	Information Model	Vision Document	Product owner	Quality, Security, Privacy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	Process Management

A drastic change in Clients' organisation was necessary to be able to deliver and manage the new application. Not only two different branches would be supported by the same application, but the support itself might also increase drastically since many new clients were expected.

The corporate strategy is not found nor references to it except the mentioning in the Vision Document that the first product in August had to be used to generate money and attract other customers. A Business Architecture, object model or information model is not present.

Quality aspects are arbitrary (not based on ISO25010) and scarcely filled. No migration plans were developed, nor a specific order in which the software should be delivered to create and steer on business value. Many artefacts are referred to, but not found. There is no overall inclusion of those artefacts (Use Case List or Business Use Cases in the Architecture). Thus, dependencies are not shown).

Security should have been mentioned since the data is valuable.

Methodology Not clear is if any and which methodologies are used to define the corporate strategy. In the Vision document, the project manager of Supplier described the strategy. Only some possible products were mentioned and a short business vision of the client. Amongst others, the EFQM model provides a framework to describe these factors but it was not used, neither are other models. BPM could serve to redesign the new organisation. This is not part of RUP either but is necessary to provide a validated business uses case model.

Contracting In Phase 0 these documents should have been delivered. The contract was not adjusted after 9-12 weeks as was agreed.

Discussion At the organisational level, the same conclusions can be drawn as at the contextual level. Supplier wrote the input for the documents, and they give no structure for the project nor addresses the real risks.

4.3.1.3 *People*

Artefacts derived from the people layer

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
People	Stakeholders needs Requirements definition	Use Case Model	Use Case Model	Product owner	Business Process Redesign HRM Training Plans, UXP, Inclusion, job satisfaction freedom from risk
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	

Use Cases describe how users will behave using the new system. These Use Cases are not sufficiently described and are not complete. The roles of employees of the buyer were only briefly mentioned in a table, and an authorisation matrix is not present. Acceptance criteria and a set up for the acceptance test by the first user are not found in any document. Training plans for the accepting user are not found. It means that input from this layer was not addressed in the Use Cases. This is particularly relevant for the service desk at Clients' office, facing substantial change.

Methodology In RUP Use Cases were developed in the Elaboration Phase, based on the business Use Cases. 50 Use Cases have been developed, and only seven of them were accepted in February 2006. The expectation was that the acceptation of all Use Cases would be ready in April 2006 (actually 9-12 weeks after signing the contract). This was called Phase 0. If Supplier meant Phase 0 to replace the Inception Phase, is not clear but in the Inception and Elaboration phase, other disciplines must also start. The first increment of the software should have been available in August 2006. This increment had to function as a basis for marketing and the user acceptance test. RUP addresses deployment and testing from the Inception phase on and builds upon that in the other phases. None of these was present except for a list of Use Cases. The acceptance tests had to start in August (but according to RUP, tests should be run after each iteration). To gain other markets and promote the product the operation of the application had to be well understood by the first acceptant.

Contracting The preparation of those documents and assisting the accepting user is not contracted. On the other hand: the contract is capacity only, thus conducting these activities should have been suggested. No evidence of that is found in the documents.

Discussion

Not addressing the people layer causes incomplete Use Cases and a lack of acceptance criteria. By not addressing the service desk governance processes are overlooked.

4.3.1.4 *Usage/Processes*

Artefacts derived from the processes/usage level

Level	IS012207		METHOD		Input from/used by
		Waterfall	RUP	Scrum	
Usage Processes	Stakeholders needs Requirements definition	Use Case Model Interaction Models	Use Case Model Supplementary specifications. Interaction Models	User stories in Product Backlog	Business Process Modelling Control/Risk: security/error protection
	Design Definition	Use Case Model Interaction Models	Use Case Model Interaction Models	User stories in Product Backlog	HRM: job descriptions and training plans Security: authorisations Control: compliance
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	

At this level, the requirements for the application, data and user functions can be completed, based on processes. These processes provide the input for the business model, Use Case model and interaction models.

An existing application was used for the requirements analysis. What this meant for the processes is not clear. In this case, the interaction model (based on the Use Case model) and the preparation of the acceptance test (acceptance criteria per business Use Case) are not found. These processes should have been completed in the Elaboration Phase.

Methodology Business Process Design or redesign is not found. Client expected the engineers to obtain the requirements from one of the existing applications. The new solution requires processes that can be tested together with the supplementary specifications, and an elaborate SAD, based on the Use Case model and interaction models. This documentation was incomplete and thus not able to guide building the system. Processes are this way not an input for the (business) Use Cases.

Contracting Process design was not mentioned in the contract. Business architects, analysts and process designers could have been delivered. Whether Client had asked for it or Supplier had suggested these capabilities is not found in the documentation.

Discussion

Client has delivered requirements by delivering the old software as input for the requirements. However, Supplier accepted it. No documentation is found that revealed different information.

4.3.1.5 *IT system*

Artefacts

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
IT	Systems/Software definition	Use Case Model Interaction Models	Use Case Model	Backlog	Operating Model, Governance, Security and Maintenance
	Architecture Definition	Information and Technology Architecture	Software Architecture Document (SAD)	User stories in Sprint Backlog	Operating Model, Governance, Security and Maintenance
	Design Definition	Use Case specifications	Use Case specifications + Additional requirements specifications	User stories in Sprint Backlog	Sprints or build teams and maintenance
	Systems Analysis	Interaction Design and DB-design	Use Case realisation, Class diagrams, Activity Diagrams. State transition diagrams	User stories in Sprint Backlog	for the above and Operations
	implementation Integration	Build, component or package	Build Implementation model	Working software	for the above and Operations
	Verification	Systems test	Test Implementation Test Results	Working software	for the above and Operations
	Transition	Production Acceptance Test	Test Implementation Test Results	Working software	for the above and Operations
	Validation	User and Business Acceptance	Test Implementation Test Results	DoD	for the above and Operations

The preparation phase should provide all documents to start the Elaboration and Construction of the solution. The documents are not signed off. The software factory is set up in January 2006 and produces software copying the old application.

In this situation the technical requirements are unclear. In the Software Architecture Document (Appendix 4.1.3) the governance of the system is hardly mentioned (except for reporting). It is not clear if the solution will be standalone or with a maintenance connection to the HQ. All ISO25010 standards were neglected. Especially the security and privacy aspects were hardly considered. The SAD contained only a 3-tier Architecture and some restrictions on security. It is not clear how the system will be hosted, operated or used. A solution for one branch is copied from an old application, while the new solution was meant to have a separate kernel and specific solutions built on that kernel. (Appendix 4.1.5, “productie 6”). This new insight could be right (although questionable) but should be at least included in the SAD. Client expects the engineers to obtain the requirements from the existing applications.

Based on the Software Development Plan (SDP) (Appendix 4.1.4) ("final draft" sic!) the production of software can start. The latest version is version 0.75 (May 2006) and is not signed off nor complete. At page 14 a kind of planning is pictured but why, what and in which order, is not described. The planning was not based on the SAD view; releases were only mentioned as a quantity of work in a given time and at a fixed budget. Which content would be delivered when and why in this order is not mentioned.

Many experts discuss the quality of the software afterwards: Client claimed that due to errors the software never functioned, Supplier did not agree: auditors hired by each party, did not agree either (Appendix 4.1.6., "productie" 12, 13 and 52).

However, the acceptance tests showed non-working software (as seen in the products mentioned before and the letters and e-mails in 4.1.7). Supplier suggested that information and instructions missed.

Supplier suggested RUP as the methodology. In the Inception Phase, a PoC can be delivered to show that a solution is feasible. In the Elaboration Phase, the Software Architecture has to be proved. The most challenging Use Cases have to be developed to be sure that the Software Architecture is functioning and the production of software in the Construction Phase hardly encounters impediments.

This methodology was hardly used: In this case, an existing other solution was rebuilt, which is not the same. It was Client who suggested that copying and rebuilding the old application was good enough at the beginning to extract the requirements. This rebuilding and use of an old application are contradictory to the idea at the start where a new architecture was proposed by Supplier: a kernel with specific functions. At least, it was unclear how this rebuilt branch specific solution would fit in the SAD-intents. The elaborate SAD, based on the Use Case model and interaction models was not ready and was never finished (Appendix 4.1.2). Any discussion about this topic could not be found. It meant that the Architecture definition, the design definition and the systems analysis was not carried out correctly. What exactly belongs to the Elaboration Phase and what to the Construction Phase is unclear. Programming went on for three years. Go-no-go decisions, based on a risk list are not found.

Contracting The contract was ended after three years of producing software. A viable product is not delivered. Client became responsible for the teams.

Discussion Serious problems arise in this area: not implementing according to a SAD, not delivering an integrated working application after three years, not even one increment. Client did not provide information on time, but still, the development teams were producing software. The only measure Supplier takes after nine months, is reaffirming the contract intentions. In the months before the reaffirmation Supplier delivered the project manager and a delivery manager. Supplier did not perform the go-no-go decisions according to RUP.

4.3.1.6 *Evaluation of the case*

The hypotheses on which the model is built are

- Projects are not delivering results because clients in steering committees lack knowledge about IT.
- The assumption is that some artefacts are more important than others and that
- The choice of methodology plays a role in this.
- ISO25010 were not used as input for the requirements.

In this case, Client did not have **knowledge** of RUP. Supplier proposed the methodology but did not act accordingly. Client did not notice.

The **crucial artefacts** are those that supported the systems development plan and the contract: the preliminary documents that should provide the structure and risk list for the project.

Supplier did not follow the **methodology** properly**.** Coding did start before the requirements/Use Cases were accepted. One could say that this is because the client stood at the steering wheel, but RUP forbids to go on if a phase is not ready. Supplier suggested the methodology and should act accordingly. On the contrary, he accelerated the programming team in India. The methodology itself is not essential: not acting according to the methodology, is.

Client did not deliver the required specifications, vision and other necessary input on time or at all. Still the project went on.

Of the ISO25010 quality requirements, hardly any is met.

Quality aspect	Findings
1 Product quality	
1.1 Functional suitability	Requirements not complete
1.2 Performance efficiency	Not mentioned in SAD
1.3 Compatibility	Not mentioned in SAD
1.4 Usability	Not mentioned in SAD
1.5 Reliability	Not mentioned in SAD
1.6 Security	Alternatives mentioned, no conclusion
1.7 Maintainability	Not mentioned
1.8 Portability	C#, no other facilities mentioned
2 Quality in use	
2.1 Effectiveness	Different opinions of auditors
2.2 Efficiency	Different opinions of auditors
2.3 Satisfaction	Not measured
2.4 Freedom from risk	Not mentioned in SAD
2.5 Context coverage	Not measured and not mentioned

Table 8

In the model, the following aspects play a role:

- **Systems have many layers (five layers for the model)**: all other layers besides the IT-system were hardly elaborated.
- **The holistic approach** is lacking which means that milestones do not address all layers
- **Intertwining with other methods**: RUP provides room by demanding a Business Case and a Business Model.

- **Initiating a project and gathering required input**. RUP also provides the initialisation phase for project management: the Inception Phase is meant for this. Each phase ends with a go-no-go-decision,
- **Contracting levels define the distribution of work**. In this case, Client was responsible.
- **More or fewer artefacts**: RUP is demanding a Vision Document, a SAD, a Business Case, a Business Model with Business Use Cases and an SDP to define a contract and start building the solution.

Summarising:

- The Vision Document was never finished, and only covered one subsystem
- The same was the case for the SAD
- For the requirements, it was accepted to copy the old system, while the new would be required to work differently and to start with another branch
- The Project Manager of Supplier never delivered a Project Plan that was signed off by both parties
- Risk management was not based on the real issue: how to generate revenue as soon as possible with this application
- Still, Supplier started with programming
- The Brinkers'-arrest shows that a supplier has to show professional conduct. Supplier did warn that documents were not ready but did not discuss pausing or stopping the project as should have been the case when using RUP.

Two questions arise:

The first question: was Client fully aware of the essence of the contract he signed?

He was entirely responsible for the steering of the activities, even if the contract mentioned that the proposed methodology was RUP. It meant that Client should have understood the consequences of going on. Supplier suggested by delivering a project manager and a methodology that they would act according to RUP, but they were ambivalent in many aspects.

Client expected a first increment to be delivered in 8 months and Supplier did not object. However, it is hardly believable that 50 Use Cases would be delivered in less than eight months, the preparation phase included.

The official letters and meetings about the interpretation of the contract suggested that Client did not understand what was expected of him. However, when he acted like the one in charge, he was cautioned not to do so. Later on, when the contract was “explained” he was supposed to be the only one in charge, having the full responsibility in steering the project.

The other question: did Supplier act according to professional standards? Supplier should not have started programming when the Inception Phase was not ready. The RUP methodology provides a gate approach: if certain documents are not ready and accepted by both parties a phase cannot be closed, and the next one cannot be started. It requires knowledge of that methodology on both sides but perseverance on the side of the one who suggested the methodology. The essential documents were never finished. Client was partly responsible for that because he did not deliver the business requirements. In reports, it was shown as a problem, not as a showstopper, which it should have been if the Supplier, according to the contract, had adequately executed the QA.

This situation provides input for further research on ethical issues.

4.3.1.7 *Answering the research questions*

The research questions we would detail here further

- Which artefacts seem crucial?
- What do project boards/clients steer/what question should/could the client ask?
- Can a model/questionnaire be derived for the client/project board? Alternatively, in this case: what should be changed to the conceptual model

What are the crucial deliverables of systems development?

According to the methodology, the documents that are necessary to create a contract at the end of the Inception Phase have been crucial. In this case: the Vision document that contains the Architecture and Business Uses Cases appeared to be essential as well as the Business Case (not present at all), The Software Architecture Document and the Software Development plan and a Risk List (not found, only a reference in the SDP). Those documents are needed to create a proper plan and acceptance criteria.

The questions the client should/could have asked

In this case, Client could have asked to explain the difference between what was expected to be ready to be able to start with the project and the current situation. It would also have been wise to have this written down in a joint document and use it as a risk list. It is not clear if and how it was discussed at all. It is a measure in RUP to create such a list.

He should have asked a clear understanding of what could have been delivered in August (and how much time it would cost after realising the Use Cases). A work breakdown structure, so that it was possible to follow, or which minimum viable product could have been ready according to usual standards, at which prices and at which milestones could have helped as guidance to steer the project.

When the contract was reaffirmed and roles were adjusted the real meaning of the contract (Supplier taking no responsibility whatsoever) had been made clear. Also, a second opinion during delivery could have been asked for by a mutually agreed auditor. In this case, many auditors were brought in by either party at the point where problems arose with payments, and they disagreed.

The peculiar situation, in this case, is that what started as performing a project, ended as delivering capacity only with Suppliers' project managers. What precisely the consequences were should have been asked (and explained).

What are the implications for the model?

An interesting finding could improve the conceptual model. The chosen methodology as such is not as important as expected: RUP gives guidance to start a project properly, it has gates that should not be skipped, and even as they did, the actual problem in the case was that according to any methodology the preparation phase for the project was insufficient. It is hard to prove in this research that the errors in the software are caused by an unstructured start (no real SAD and SDP), but in other studies, the relation between the quality of the requirements and the specification leading to quality (acceptance) criteria is already proven (J.F. Groote, 2014).

Knowing what and why should be present exactly before starting to deliver an IT system is more important. Maybe it even would have led to another project by discussing alternatives first. Where calculating the costs on the alternatives and the possibility to create revenue were the base for working together. These alternatives were not considered, which was especially crucial in this case to create revenue. It might have led to earlier termination of the contract.

In this case, the contract is the most important. The steering of both parties and steering by Client was not distinguished well.

The analysis of this project according to the conceptual model shows the importance of considering the whole system, from supra-system to IT. Contracting is a good starting point in terms of customer expectation. In this case, the rigour of the methodology could have given support to underpin this situation.

This case also shows that the Project Plan (SDP) needs to cover all these insights. The content of the project plan matters. It is a necessary document to understand the cooperation between the parties and to understand the real risks (in this case, not generating money on time). When this content is related to the appointments in the contract and understood by both parties, reporting makes sense.

A preparation phase to get a clear starting point to agree upon a contract and deliver according to it is a necessity. Moreover, open communication between both parties also is.

4.3.2 The Public project

The start This Public project was another version of a system that had been developed in the years 1992-1994. It delivers allowances for healthcare.

The reason the public system was restructured, was that parliament wanted to diminish fraud by the users of the system. The known fraud rose rapidly from 1,5 M€ in 2013 to 4-5 M€ in 2014. Causes were unclear. The assumption was that users frauded. This assumption was the leading principle for the business requirements of the system (Appendix 4.2.1).

The project order was officially dated on the 12th of June 2013 and a roadmap was returned at the 3rd of July 2013. The legal underpinning of the requirements came into effect on the 3rd of December 2014, parliament discussed it in April 2014.

The context of the system was changing drastically, and the new supra system had to be functioning at the 1st of January 2015 as the new law came into effect. Two other legal tracks (Care Insurance Companies and Permanent Care) are left out in this chapter to improve the readability. These tracks encountered problems, but since their contextual system changed less, the situation could be handled.

A government institution developed the system, residing under another ministry than the health ministry (from now on this institution is referred to as ZBO (independent government entity)}.

Another way of working was introduced with the new law. Instead of getting a lump sum personally to spend on care, the budget was granted (or not) by the municipalities, the municipalities would judge the contracts, and the ZBO would manage the budget and pay the care providers according to their bills (underpinned by contracts).

Old situation	New situation
Professional assessors determined the budget and sent it to ZBO	Three possibilities, one of them via municipality. In that case assessment by a civil servant. Rules per municipality differ
Budget was handed over to the client by ZBO	When determined ZBO gets information of municipality, and assigns a budget
Client contracted (in some cases ZBO pay-rolled)	All contracts are sent to ZBO, contracts/care provider is checked
Client paid care providers	ZBO checks amount/contract and if ok pays bill until maximum
Incidental checks and audits	Check on validity when bill arrives

135.000 clients, 700.000-1.000.000 contracts had to be processed for the first time and 250.000 payments per month. The result was that the implementation of the system costed 156 million, the payback time for preventing the deciphered fraud is therefore 40 years.

Contracting the work The ZBO delivered a project plan twice. The first in 2013 after the order in a ministerial letter. The legislation changed during the execution of the project and a second plan was delivered in July 2014, a half year before the due date. In this plan lots of remarks were made about the uncertainty of legislation, the impossibilities of specific solutions and questions were asked. However, this was not a formal letter from the board of directors (BoD), and the project was expected to deliver at the first of January.

A formal letter was written by the BoD to express their unhappiness about the changing requirements and the lacking chain architecture. (Appendix 4.2.2.). The content was discussed in August, four months before the due date.

The execution The project chose for Scrum as methodology. PoC's were executed with municipalities and bodies of interest before the care system was again changed.

Not clear is how Scrum was implemented: the project did not deliver software per sprint. At some point the execution was different from the plan: the former applications were used, and only a sunny-day-scenario was developed, indicating progress was not turning out as intended.

In the meantime, the care system changed drastically and was amended at the same time by parliament. The change was not meant as cutting on budgets but caused in this area; a cut down to 50 %, which caused lots of questions and objections.

When the deployment started it did not turn out as promised:

The service-desk was not equipped to handle the number of questions. In a short period, 1000 extra service-desk employees were hired, and a team handled the most urgent cases. After a few months, all cases were urgent since some care providers did not receive money for months. The ministry decided to pay all bills without checking. This period took about a year. Many measures were taken, and a new system had to be built: the new project plans were twice reviewed and considered weak by BIT (Bureau of ICT reviews of the government) especially regarding the (lack of) collaboration of all partners in the chains.

Main reasons for this failure were:

- a tracking possibility was not available, so the ZBO actually handled questions of the municipalities
- clients lacked digital skills and the solution was not built accessible.
- The new payment system was a drastic change
- Lack of enough trained and skilled civil servants of the municipalities.
- A poor designed user interface for the service desk made them unable to share more information as the client already possessed. A Rapid Response Team handled the severe cases (at first 25 and later 50 employees) and got extra information.

The fraud did not end. This system hardly detected malicious care providers. It turned out that they caused most of the fraud. These fraud cases were detected by police work. In the first year, all controls were removed, to accelerate the handling, which caused new possibilities.

4.3.2.1 *Contextual Level*

Artefacts derived from the contextual level

Level	IS012207	METHOD			Input from/used by
		Waterfall	**RUP**	**Scrum**	
Contextual	Business and Mission Analysis	Definition study Business Object Model Architecture Vision	Vision Document Business Case	Product owner	Marketing and/or Public Affairs/Legal Board of directors
	Stakeholders needs Requirements definition	Requirements model	Vision Document, Business Use Case Model	Product owner	Marketing and/or Public Affairs/Legal Finance & Control
	Systems/Software requirements	Requirements model	Vision Document	Product owner	All stakeholders for this level
	Architecture Definition	Architecture Requirements	Software Architecture Document (SAD)	Product owner	All stakeholders for this level

In NORA (the Dutch Government Reference Architecture) architecture is prescribed for each project. Nora also contains a recommendation to implement a case-system in order to track and trace the handling of the case by a government institution. The contextual Architecture for the change was not delivered, not by the ZBO nor by the ministry (Appendix 4.2.2). The absence of Architecture was discussed in August, four months before the due date. The process design appeared to be lacking provisions for business continuity in the chain (since a sunny day does not require it).

Methodology No overall architecture, especially not for the interactions in the chains, could be found. Intentions are made to use existing paths. At many points, decisions have to be made but are not enforced. Complicated is that the cases should provide insight into the status of multiple institutions.

Public Affairs and influencing the decisions of Parliament was difficult. The plan of July 2014 (Appendix 4.2.7, page 9) showed still unsolved issues. The analysis of the changes in the old legislation and new legislation and the feasibility study is not found. Until November the ZBO insisted that a working system would be delivered. The Swimming lanes in the Project Plan provide insight, but the choice of the sunny-day-scenario is not underpinned: 90% of all requests appeared to be non-standard. Also, the other partners in the chain needed handling not at least the care insurance companies.

Contracting One of the problems appeared to be the governance structure: who exactly was the client and supplier (contracting partner): Was ZBO supplier to the Ministry of the application or was it the Ministry of the ZBO? ? Moreover, what exactly was the role of the municipalities and their umbrella organisation and the Care Offices? A written document in which all roles were clarified is not found. On the other hand, in the Plan the ZBO would cover everything by using Scrum.

Discussion The absence of a contextual architecture (with a thorough as-is-to-be analysis) caused that no attention was given to the new interactions between the institutions. To three stakeholders new applications were delivered: at the Care provider, the Care Clients, the Municipalities and the ZBO itself. An explanation of choice for a sunny-day-scenario as MVP is not found. Scenario planning is not found at all.

4.3.2.2 *Enterprise Level*

Artefacts derived from the enterprise level

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
Enterprise	Input upper level	Business Use Case Model Business Architecture	Business Use Case Model	Product owner	Corporate Strategy Product management (product and services) Business Process Improvement
	Design Definition	Use Case Model	Use Case Model	Product owner	Planning & Control, KPI's Business Process Redesign
	Design Definition	Information Model	Vision Document	Product owner	Quality, Security, Privacy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	Process Management

It is unclear if the product owner had to handle only the ZBO or also all other parties in the chain or how this product ownership across the chain was organised. Which products or values streams have been recognised is derivable from the PoC's that have been tested: a portal for municipalities and the bodies of clients. PoC's for care providers and other institutions are not found.

The Architectures according to NORA are not found, the operating model neither. A process design is not found. Especially the interaction with the service desk is not seen.

Methodology A Process/Business Architecture was lacking only working instructions were available. Swimming lanes in the Project Plan showed that a process design was foreseen, but the interaction between the municipalities and ZBO was not mentioned, and the cohesion with the user stories is not found (Appendix 4.2.7, page 32,33). Product 6 shows that processes are described after the introduction. The last Plan of July 2014 (sic!) shows fragmented topics. Scrum is used "to build the functionality we know. Otherwise, we would have to wait for the requirements to be ready". In the plan, no references were made to other studies.

Contracting The project plan served as the contract. It changed several times since legislation and plans of the Client changed. In the plan, many questions were asked, and opinions were given, but the consequences in time, money and unfinished functionalities not. All stakeholders resided under a different Ministry: the ZBO, the care and the municipalities. It seemed that the ZBO was responsible to deliver the total solution but had no power to enforce decisions.

Discussion The process designs were not based on the contextual level. Business continuity was not designed. The change in all other systems was not managed. Unclear was who was responsible for what. A good (Business of Enterprise) Architecture is lacking. Especially the choice of the sunny day scenario is missing. It seems like an emergency. The risk list to underpin that choice is missing in the Project Plan.

4.3.2.3 *People*

Artefacts

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
People	Stakeholders needs Requirements definition	Use Case Model	Use Case Model	Product owner	Business Process Redesign HRM Training Plans, UXP, Inclusion, job satisfaction freedom from risk
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	

In this case, many People categories play a role:

- The clients: in need of usable and accessible software and instructions since many people have physical disabilities.
- The care providers: many providers did not receive their wages and did not know how to handle this situation. The client was responsible for their payments
- Civil servants of the municipalities: new legislation, a new function in a new care system.
- Service desk: answering the questions for all users.

Acceptance tests were not performed. The PoC's served as such

Methodology The ministry did not use standard methodologies to adapt people to change. The target group is somewhat unique, with specific problems. The change was not absorbed. The political reality and the time people needed was not in sync. No visible action was taken. Actually, in government, the principal value is equal treatment and rights in the same circumstances for all citizens. This value was not the leading principle in delivery: they delivered a sunny-day-scenario, which implies unequal treatment.

Contracting It is unclear who had to instruct the clients and the care providers. The ZBO was responsible for delivering a service desk. The municipalities had no trained staff to service this process,

Discussion The vulnerability of the target group was underestimated.

At the municipalities no sufficient and trained staff was available.

To instruct the clients and the municipalities the software should have been available much earlier.

4.3.2.4 *Usage/Processes*

Artefacts derived from processes/usage

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
Usage Processes	Stakeholders needs Requirements definition	Use Case Model Interaction Models	Use Case Model Supplementary specifications Interaction Models	User stories in Product Backlog	Business Process Modelling Control/Risk: security/error protection
	Design Definition	Use Case Model Interaction Models	Use Case Model Interaction Models	User stories in Product Backlog	HRM: job descriptions and training plans Security: authorisations Control: compliance
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	

The processes, as already mentioned above are not explicitly designed and did not serve as a guiding principle for the user stories. Existing applications are introduced to interact with the newly developed portals (according to Appendix 3.2.3), but the interfaces with them are lacking. In this case, the Use Case model, interaction model and the preparation of the acceptance test (acceptance criteria per business Use Case) are not found. Probably documentation was present in scrum teams, but task forces were afterwards restructuring and producing documentation to prove that the processes and applications were compliant to the legislation.

Methodology The process was designed around the sunny day scenario, different from the documented Project Plan. In the documentation, no process descriptions are found except for grouped user stories. Which processes were crucial to start with is not determined. In reality, the client could not verify the status of his nor the amount of money left.

Contracting It is unclear whether processes are part of the assignment. However, the internal processes are the responsibility of the ZBO.

Discussion At this level new entities are introduced: the old systems are used as a base for the new, while they were functioning differently. Not the new process design but tying together the old systems with the new portals became dominant. The process was developed for each entity apart (Municipality, ZBO, Client) but not as a whole integrated system. Why the internal processes were not developed (service desk, business continuity) is not found. Afterwards, processes have been developed by several task forces.

4.3.2.5 *IT system*

Artefacts

Level	IS012207	METHOD			Input from/used by
		Waterfall	RUP	Scrum	
IT	Systems/Software definition	Use Case Model Interaction Models	Use Case Model	Backlog	Operating Model, Governance, Security and Maintenance
	Architecture Definition	Information and Technology Architecture	Software Architecture Document (SAD)	User stories in Sprint Backlog	Operating Model, Governance, Security and Maintenance
	Design Definition	Use Case specifications	Use Case specifications + Additional requirements specifications	User stories in Sprint Backlog	Sprints or build teams and maintenance
	Systems Analysis	Interaction Design and DB-design	Use Case realisation, Class diagrams, Activity Diagrams. State transition diagrams	User stories in Sprint Backlog	for the above and Operations
	implementation Integration	Build, component or package	Build Implementation model	Working software	for the above and Operations
	Verification	Systems test	Test Implementation Test Results	Working software	for the above and Operations
	Transition	Production Acceptance Test	Test Implementation Test Results	Working software	for the above and Operations
	Validation	User and Business Acceptance	Test Implementation Test Results	DoD	for the above and Operations

A sufficient Software Architecture based on the application and chain Architecture was missing. Especially the use of old systems (T and N) was not mentioned. It seems they delivered portals (client, care provider, municipality) without coherence, based on applications that were developed for another purpose. An integration model was not found, systems analysis is not present as if one of the agile principles (“working software over comprehensive documentation” from the Agile manifesto was understood as hardly any documentation,

A logical architecture of the system, on which the development did start was not available. The picture in the Project Plan (product 7) only shows very high-level planning (page 27). A first increment of the whole system was delivered instead of an MVP.

Also, the XML, used for the exchange of data with municipalities, could not be implemented by many municipalities, because they used the same software supplier. The specifications came too late to build the new interfaces.

The ISO25010 requirements were hardly met: The Project Plan mentioned how many users had been calculated. Maximum use (when contracts should be renewed), payments (totally 250.000 (in reality it was 750.000), business continuity (what if...) was not developed: a sunny day scenario does not require this, but it can cause unfit solution (product 5). Because the websites were often unavailable, sizing could not be calculated well.

Methodology The ZBO used Scrum. In the Project Plan these phrases can be found The ZBO "heeft bewust gekozen voor de Scrum methode in dit traject. Bij een traditionele ontwikkelmethode (waterval) zouden eerst alle specificaties moeten zijn vastgesteld waarna het ontwerp en de bouw zou volgen. Dit zou betekenen dat nu nog niet zou kunnen worden gestart omdat de scope nog niet is vastgesteld. Door te kiezen voor de scrum methode kon reeds worden gestart met die zaken waarvan al bekend was dat die sowieso gebouwd moesten worden" (product 7) and means: we used scrum because we did know exactly everything and started with the parts of which we knew they had to be built.

Essential in Scrum is to deliver usable software every sprint. This was not managed, nor was this way of working mentioned in the Project plan.

Moreover, last but not least: even Scrum needs well-defined requirements. The product owner should take care of that. The letters that were sent to the Client to get more detail are sent rather late in the process (four months before delivery-date).

In this case too, programming had already started, and former applications were used as a base. These applications worked according to other principles. An Application and Software Architecture is lacking.

Contracting

The discussion started in the project plan, a half year before the due date is a way of negotiating the contract. It led to external audits and reviews to decide what was needed to get the work done.

The use of the old systems was not mentioned in the plans. Not clear is if the provided money was meant for a new system. The "contract" did not mention that old software was used. A real contract was not present, but the Project Plan served as such. Because the Architecture was not available, guidelines for the IS system were not available on time.

Discussion At this level the product owner should provide all necessary input. It prerequires a SAD to know in which order. The project plan shows that the use of Scrum is not as intended. No delivery in value streams and no shippable software per sprint.

Lots of evaluations and audits were executed afterwards, and it became evident that the software could not support the system.

Quality assurance was performed as a standard procedure (gateway reviews) but not regularly. The quality was only assessed as problems arose. The assessments covered a specific question or problem.

4.3.2.6 *Evaluation of the case*

The hypotheses on which the model is built are

- Projects are not delivering results because clients in steering committees lack knowledge about IT.
- The assumption is that some artefacts are more important than others and that.
- The choice of methodology plays a role in this.
- ISO25010 is no input for the requirements.

In this case, the ZBO does not act according to Scrum and delivers too little documentation. The late discovery that the project could fail suggests that **knowledge** about Scrum is lacking. The project was not steered according to Scrum. The Client and probably the board of directors maintain optimistic until November.

The **crucial artefacts** are those that support the systems development plan and the contract: the preliminary documents that provide the structure and risk list for the project. Especially in this case the contextual architecture.

The ZBO did not follow the **methodology** properly**.** Coding was started before the requirements/Use Cases were ready. It was even presented as an advantage of using Scrum. The methodology itself is not essential: not acting according to the methodology, is again.

The ZBO asked for the specific requirements only in July, a half year before the due date. This left no time for restructuring. It looked as if using the old software was a measure to accelerate the realisation.

Of the ISO25010 quality requirements, hardly any is met.

Quality aspect	Findings
1 Product quality	
1.1 Functional suitability	Requirements not complete
1.2 Performance efficiency	Turned out to be negative
1.3 Compatibility	Other parties were not able to deliver
1.4 Usability	Not present for this target group
1.5 Reliability	Not mentioned
1.6 Security	Some mentioned, a DPIA is executed, but the outcome is indecisive
1.7 Maintainability	Not mentioned
1.8 Portability	Not known
2 Quality in use	
2.1 Effectiveness	It did not function
2.2 Efficiency	Social and parliamentary outrage
2.3 Satisfaction	See above
2.4 Freedom from risk	Not known
2.5 Context coverage	Not sufficient

In the model, the following aspects play a role:

- **Systems have many layers (five layers for the model)**: an inclination for the IT-layer was present other layers were neglected.
- **The holistic approach** is lacking which means that milestones do not address all layers
- **Intertwining with other methods**: the product owner needs to know what is needed. The methodology does not give guiding principles.
- **Initiating a project and gathering required input**. See above. Scrum does not guide the product owner or scrum master to behave efficiently.

- **Contracting levels define the distribution of work**. In this case, the ZBO seems responsible.
- **More or fewer artefacts:** documentation was hardly produced. Scrum gives no guidance for that.

As in the previous case, the governance and agreements are not understood the same by both parties at the start of the project and when problems arise this becomes an issue. No formal contract is present, but an explicit agreement on the specific terms of delivery is not made either. The role of the two ministries is unclear, and it is not clear what exactly should be delivered by each party. Actually a third ministry is involved as well (under which the municipalities reside): disagreements on the scope and content were present until the implementation (and after): in the end the scope was perceived by the ZBO, as only delivering an IT-system according to specifications of the Client, but these specifications were lacking and delivered in a very late stadium. The ZBO could have asked earlier, more explicitly and made the consequences clear. However, the official reports until November mentioned expecting delivery on time.

Things went obvious wrong and led to resigning of members of the board of the ZBO. What were the main root causes?

- No real agreement on who is performing which task
- Not applying the government principles of Architecture
- A strong focus on using Scrum instead of designing an integrated system on all levels
- A change of plans by introducing the sunny day scenario and the use of the former systems without communicating the effects of these choices
- Underestimating the vulnerability of the target group
- Underestimating the readiness of the municipalities.
- Overestimating individual fraud and underestimating fraud by malicious care providers.

4.3.2.7 *Answering the research questions*

The remaining research question to be answered in detail:

- Which artefacts seem crucial?
- What do project boards/clients steer/what question should/could the client ask? And
- Can a model/questionnaire be derived for the client/project board? Alternatively, in this case: what should be changed to the conceptual model

What are the crucial deliverables of systems development?

The agreement in which is described who is responsible for what should have been the leading document in this case, because this was the only base for reporting to the Client. Scrum is a methodology that does not deliver fixed content at fixed dates, but delivers according to value streams in Sprints: the question is then: what should have been the right value streams and would it be delivered when needed with the right quality or was the time path too short? In the political context, it was hardly possible to tell beforehand that the planning could not be met, but in hindsight, this outcome was worse.

Another crucial element is the handling of the progress reports. Until November was reported that the complete system would be delivered on time (not only the “sunny-day-scenario”).

In this situation it is quite clear that the system is more than an IT system: the lack of design of the external interactions, processes and the impact analysis have been found crucial.

When designing the processes of the system first, most problems could have been prevented, and the systems could have followed another implementation path. Not clear is if the 1st of January would have been reached, which was very important.

What should/could the Client have asked?

Also, in this case, it is crucial that Client and Supplier understand and agree who is performing what to deliver a successful joint program. It could be possible that Client trusted the outcome since no problems were reported until November. The Client should have asked to get insight into what is delivered when for which stakeholder and verify if stakeholders were able to work with the solutions timely. In this situation no slack time was available. By providing an architecture, the plan could have been tracked against that architecture. Now the delivery could not be measured

What are the implications for the model?

For the conceptual model, it is essential that contracting (in this case informal) has to be clarified. Scrum does not give much guidance on what should be prepared to create a good backlog. However, the methodology is not used as meant as well. Again, an unstructured start led to a negative outcome.

Knowing what and why should be present exactly before starting to deliver an IT system is more important. The input of all layers cannot be neglected and has to be structured in a model.

4.3.3 The preliminary implications of the case studies for the model

The two case studies showed many similarities:

- The understanding of who was responsible for what was not clear.
- Although the requirements and Architecture were not ready, programming in both cases started. In the Public case, it was even explained away with the argument that with the use of Scrum this was possible and accelerated the project.
- The analysis of the requirements was in both cases in hindsight not thorough enough: In both cases, decisions were made as if they were inevitable. Important principles (and thus crucial for risk planning) were not leading: selling the application to generate revenue in the first case and a system that guaranteed equal treatment of the clients in similar circumstances in the second case.
- In both cases, these real risks (that actually took place) were not in the risk list. The risks on other levels than IT were not or hardly mentioned and were no steering principle.
- In both cases many problems arose outside the IT realm, they influenced the outcome but were neglected or not handled (the enterprise level in the Private and the contextual level in the Public case)
- In both cases, the methodology was not well understood by the client. In both cases, the supplier did not act according to the methodology, and the client did not notice.
- A sixth similarity is that both clients got progress reports that reported on some issues, but they were not marked as showstoppers. Progress was made according to the reports and smileys were smiling.
- The Project Plan was not based on the mutual understanding of which activities had to be carried out by whom.
- The other deliverables seem less relevant: even being insufficient, and leading to incorrect outcomes, this was mainly caused by the incomplete preparation phase since in both cases requirements and Architecture (even Software Architecture) are not delivered before starting programming. In small projects, a strong vision and goal setting could replace it, if the architecture is not too complicated.

These findings lead to an adjustment of the model:

- The mutual understanding of who is performing which tasks per system level in the preparation phase should be added;
- Contracting the delivery is an outcome of the preparation, not the start.
- The methodology is only important in following it as intended, and thus included in the mutual task list.
- All levels must be considered to create success.
- ISO25010 are just standard requirements. However, if neglected problems occur.

4.4 Justify by a survey

4.4.1 The survey

To evaluate and detail the model with the relevant business needs a survey was conducted amongst decision makers in project boards (of distinct projects) in order to understand if there is a difference between well-run projects and failed projects. In total 38 completed the survey. The following hypotheses for the successful completion of a program/project were tested:

Via SurveyMonkey, the survey was accessible and completed from 11-5-2018 until 8-7-2018.

Thirty-eight people answered the questions (and thus data of 38 distinct projects were used). This amount is not enough to perform statistical analyses, but it provides valuable information for the assumptions on which the model is based.

The survey contained 21 multiple-choice questions, sometimes with the possibility to explain or to add a not prescribed answer. All questions were answered.

The survey was spread via e-mail, and LinkedIn personal messaging especially to senior responsible owners as Prince2 calls them (business owners, problem owners) and other project board members but also program and project managers.

4.4.2 Content and intent of the survey

The survey contains questions on the following topics (see Appendix 3)

- the organisation, the role of the answerer and the project (1, 3, 4, 8, 9, 21),
- contracting, costs and governance (4, 10, 11, 13, 18),
- the results and the Business Case (3, 4, 5, 6, 7, 13)
- questions about methodology (14, 15, 16, 18, 19, 20)
- quality and content (12, 14, 15, 16, 17, 18, 19, 20)

The intention was to see what business owners and others who are accountable for the results of projects and programs know about their projects. The coherence between the given answers in a methodological way is more important than the outcome of the projects. However, it is examnined if there are any relations.

In this survey the following hypotheses were tested:

- Programs/projects that are successful have a clear goal as a driver for the Business Case that can be described by the business owner (6, 7, 8).
- The constituent elements of the Business Case are known (4, 5, 6, 7, 8, 13).
- To deliver the Business Case processes should be known/developed before programming and alternatives should be considered (14, 15, 16).
- For customised development ISO25010 norms are considered beforehand to deliver quality at once to control costs (11).
- Quality management should be present and positioned outside the program/project; otherwise, the business owner cannot get a clear view of the results (12).
- Methodology matters and relations between methodology and activities are crucial (2, 14, 15, 16, 17, 18, 19, 20).
- Each type of contract needs its own preparation up front (4, 6, 15, 17, 20, 21/28).

4.4.3 Testing the hypotheses

To demonstrate the way we used the outcome of this survey, an example is given of the completion of projects corresponding to the methodology used:

Only 7 out of the 38 projects/programs are completed. The others are still running: 19 are partially implemented, and 12 are not implemented at all. Crucial in agile environments is that minimum viable products and improvements are implemented and used. Nine projects from the ones that had no delivery at all were delivering according to agile methodologies. That would mean that they had to deliver intermediate results.

If we dive further into the figures, we find out that:

- Three projects from those nine not delivering at all until now are planned for only one year but burning over 5M€ for IT delivery in less than a year. Two of them are producing customised software. These figures do not correlate. In these cases, the respondent is not able to deliver a description of the purpose of the project. In one case the results are already over budget, and the goal will not be reached. In the other everything is considered as good, but the quality assurance is executed in the project. It can be possible that they deliver at once, but this is rather “un-agile”.
- Two projects take longer than three years, without delivering results yet, which makes it hard to believe that this is agile.
- Four projects have a duration between one and three years without delivering any product. This is especially strange while three projects have a budget larger than 5 million euro.

Of course, these relations are statistically not significant enough, because of the small numbers, but 6 out of 9 projects are certainly questionable, if not 8 out of 9. Moreover, that should disturb the business owners.

4.4.3.1 *Programs/projects that are successful have a clear goal as a driver for the Business Case that can be described by the business owner (6, 7, 8)*

All completed projects (7) have in common that they delivered within budget even when one of them would not deliver results at all. Only one of them reached the aimed results entirely, and the other five claim to reach the results partially. This is contradictory to the fact that they claim that they delivered within budget. Three projects used agile methodology the others did not. All decided on scenario's beforehand.

Other interesting findings are: time and money are relatively limited in these cases: the duration of two projects is less than one year, four less than three years and only one longer than three years. The budget of three projects is less than 1 M€, of two projects less than 5M€ and only one over 5 million euro.

If we define success as having reached the goals instead of ready, we see similar correlations, but also some interesting differences: when the goalsetting has been executed before the project started the realisation of the budget is also positive. However: only one of the projects is implemented. Facts do not underpin this positive mindset.

4.4.3.2 *The constituent elements of the Business Case are known (4, 5, 6, 7, 8, 13)*

The most important part of the Business Case is knowing what effect must be achieved: this goal-setting is the guideline for the project when choices have to be made. When these goals are not defined beforehand the results of the projects are undefined: in these five cases, three are exceeding budget, and none of them is implemented.

Where the goalsetting is executed, only 5 out of 33 projects have not been implemented and do not exceed budget, only 3 out of 33 are not reaching their goals.

4.4.3.3 *To deliver the Business Case processes and Architecture should be known/developed before programming and alternatives should be considered (14, 15, 16)*

All completed projects (7) have in common that they considered alternative scenarios beforehand. On the other hand: only defining alternative scenarios is not enough: 8 out of the 27 that considered scenarios were not yet delivering anything. However, in 4 of those 8 cases, the respondents had not defined the goal of the project. In three cases the respondents claim the goals are not reached.

In 23 cases the process definitions are delivered before programming started. Only two cases did not deliver according to the intended goals. Seven of them reached their goals fully. Which is not the same as completed (only four were). When we consider the cases in which the processes were not made beforehand, the compliance to the goals seems almost equal, which is an unexpected finding.

Architecture is delivered in 27 cases before the project started. There seems no significant difference or correlation if the Architecture is made beforehand, during the planning or afterwards. This is contradictory to the findings of Slot (2010). However, in this survey, many projects have not been finished yet.

4.4.3.4 *For customised development ISO25010 norms are considered beforehand to deliver quality at once to control costs*

The cases with the highest score on the ISO25010 norms were as costly as the full range. 30% exceeds budget (and even 16% exceeded budget over 50%). A possible answer to this could be: the other projects will notice in a later phase, or this makes projects more expensive than expected because now only the sunny day scenario is developed, but these are educated guesses: all projects need to be completed to make a full comparison and application management afterwards has to be measured.

Interesting is that reaching the goals entirely, is significantly higher: 38% versus 23% for the whole population. A possible explanation could be that implementing these factors need more analysis beforehand and discussion on how and what.

4.4.3.5 *Quality management should be present and positioned outside program/project (12)*

The hypothesis is that quality management should work for the business owner to be independent of the program, in order to get full information about the project.

Only in 39% of all cases independent quality management (QM) is present. However, in 87% of these cases products have already been implemented. This percentage is significantly higher than 57% of the projects which did not yet deliver. In only 20% of these cases budget was exceeded (instead of the average 50%).

4.4.3.6 *Methodology matters and relations between methodology and activities are, and each type of contract needs its specific preparation up front (14, 15, 16, 17, 18, 19, 20)*

In only six cases the preparation phase was executed as intended and 67% of these cases delivered results. However, 50% of the respondents of these cases were the project or program managers. A bias in completing the survey could be possible.

It is notable that organisations themselves steer 27 out of 38 (71%) projects (56% with a project leader from the organisation itself, and 35% with a freelancer). Of all 27 projects six (22,2%) delivered totally according to the purpose and of those who were steered internally 25%. Just one did not deliver at all. 33% exceeded budgets of which 67% over 50% (22 projects).

Delivering according to specifications is only the case for four projects. Reaching the purpose is also 25%, and none is expected not to deliver. However, all these projects were completed within the limits of 20% of the budget. Three out of those four are deployed.

"Delivering for use" is the case for four projects: all projects delivered a positive value but none according to the original intentions. In one case budgets were exceeded (25%). Notable, but statistically not relevant, is that that is the one where scenarios were not discussed beforehand.

4.4.3.7 *Other questions*

The relation between the project manager and the board: In 18 cases (47%) the project manager is an employee, in 37% of the cases the project manager is a freelancer, explicitly hired for this project. This means that in only 15% of the cases the project leader is hired of a consultancy firm.

The projects from the internal managers have been reaching the goals wholly (28%) or partially (72%).

The freelancers reached a comparable score: 21% fully, 71% partial and 8% not.

In the other population (only 6) 50% did not reach goals and exceeded budgets. However, the population is relatively small.

In only two cases delivery occurred as fitness for purpose. The two have different outcomes (one positive and one negative). In the negative case, alternatives were not discussed, and the budget was exceeded. A bit contradictory in those cases is that the steering group chose the methodology, especially while one of them was implementing networks.

4.4.4 Discussion of the results and implications for the model

Interesting is that the answers on the projects that have not delivered yet, are given based on expectations and could have a bias towards success. Especially about the purpose: in almost all cases a strong belief exists that the original goals will be met, and if not all, the Business Case will still be significantly positive. However, the budget is already overrun in 29% of the cases and not known is what the not completed projects will do in the end.

Even with these remarks, there are some notable results:

1. Former methodologies as RUP, DSDM (also Agile) are not used (maybe in company manuals that rather often turned out to be used).
2. Insiders of the organisation populate many of the projects and programs, sometimes with a hired professional as program manager, without contracting about responsibility: However: those projects that have been executed as fixed price projects all deliver within budget with a slightly better outcome.
3. Defining goals and considering alternatives before starting a project seem crucial for successful delivery, it also causes that projects run within budget.
4. Considering ISO25010 quality aspects does not seem relevant.

The first result was not expected, but it raises the question of whether project boards are willing to change their organisation in an agile organisation or that agile promotors influence them without understanding the consequences. The combination of delivered results and the time spent does not fit the Agile way-of-working.

The second result shows that organisations choose to conduct projects themselves instead of a large software company running the project.

That common goalsetting beforehand led to successful projects was the other side of the coin of the case studies, and expected, but now reinforced.

The survey underlines that methodology is not the key, but preparation before starting a project is.

For the model it means that these outcomes endorse the proposed adjustments in the previous sector: the preparation before a project starts seems crucial. This means that goal setting will be emphasised.

4.5 Evaluate: Reflection and Implications

The justification consisted of two different approaches:

- The analysis of two case studies, that very visible failed. Although very different on first sight, showing the same problem: Client and Supplier failed to understand that they did not have got a reciprocal agreement about who had to perform which task at each level of the whole system. A strong focus on the IT system was present in both cases. All other system layers were poorly addressed.

- The survey was meant to examine successful projects. 38 project board members were asked to take in mind a completed project. However: many projects were not completed at all. This could demonstrate a "governors' bias", i.e. their belief that results will be met, even when indicators point in another direction. This bias questions the accuracy of the answers. To perform an in-depth analysis of these projects is out of scope of this research.

Interesting is that the answers on the projects that have not delivered yet, are given based on expectations and could have a bias towards success. Especially about the purpose: in almost all cases a strong belief exists that the original goals will be met, and if not all, the Business Case will still be significantly positive. However, the budget is already overrun in 29% of the cases and not known is what the not completed projects will do in the end.

Even with these remarks, there are some notable results:

1. Former methodologies as RUP, DSDM (also Agile) are not used (maybe in company manuals that rather often turned out to be used).
2. Insiders of the organisation populate many of the projects and programs, sometimes with a hired professional as program manager, without contracting about responsibility: However: those projects that have been executed as fixed price projects all deliver within budget with a slightly better outcome.
3. Defining goals and considering alternatives before starting a project seem crucial for successful delivery, it also causes that projects run within budget.
4. Considering ISO25010 quality aspects does not seem relevant.

The first result was not expected, but it raises the question of whether project boards are willing to change their organisation in an agile organisation or that agile promotors influence them without understanding the consequences. The combination of delivered results and the time spent does not fit the Agile way-of-working.

The second result shows that organisations choose to conduct projects themselves instead of a large software company running the project.

This section consists of two parts:

- the reflection, which discusses the answers to the questions:
    - Which artefacts seem crucial?
    - What do project boards/clients steer/what question should/could the client ask?
- The implications on the model/questionnaire for the client/project board:
    - What should be changed to the conceptual model.

4.5.1 Reflection

Crucial artefacts

To answer the questions on the artefacts, it is important to analyse the problematic as well as the successful projects and decide which artefacts are related to them according the theories of system development. This gives an indication of the crucial artefacts.

In the order of importance, they are:

The cases as well as the survey show that goal setting (or the absence of a thorough goal setting and analysis) is important.

Goal setting is part of the mission and vision analysis and stakeholder requirements definition and requirements analysis on a high level. Artefacts of these processes are the requirements definition, the mission and vision analysis and the to-be-enterprise architecture.

The constituent elements of the Business Case are known

The most important part of the Business Case is knowing what effect must be achieved: this goal setting is the guideline for the project, and it resembles the first. Knowing which products and services contribute to the result give focus and provides a better risk list.

Defining goals and considering alternatives before starting a project seem crucial for successful delivery, it also causes that projects run within budget.

Considering alternatives is part of the highest level of architecture: defining the difference between the as is and the situation to be and finding the most promising paths and valuable paths. When defining the architecture further in software architecture the same exercise applies to smaller IT only-system-parts. In both cases this was not carried out properly and the survey shows a slightly better result when these actions were executed.

All levels of the system have to be addressed

In the cases it became quite clear that, when system levels are neglected problems will occur at the IT-level which cannot be solved there. The survey provides an indication for that, however less clear.

Project reporting

In the cases it was seen that the progress reports contained sunnier expectations, than logically should be the case. This resembles the discrepancy in the survey in which the indicators show worse results than the expectation of the business owner.

Questions.

The questions, in general, the client should or could have asked based on the findings are based on the preliminary work:

- What do I get, in which order and can you prove that it is the right order or: How do you guarantee that this creates the best value? With which methodology is not important, the result is.
- What do you expect from my organisation, when?

This seems very obvious but in both cases these questions were not transparently discussed between Client and Supplier.

This meant that during the projects, although the projects were in bad weather, the project reports did not resemble the situation and the client was not able to recognize this. By asking the same questions as above and checking if the results are met, this can improve.

Conclusion

Discussing these findings:

- The methodology is only essential because some methods do not guide the preliminary phase. This gap has to be bridged and cannot be overlooked.
- To know how to cooperate, a mutual agreement is necessary. it should not only be clear who is responsible for managing the outcome at each level of the system. Also, the specific task list is needed as input for a plan to ensure that all activities are completed when the project goes live. This means that the plan must show the interdependencies between those activities and milestones.

4.5.2 The preliminary implications of the case studies for the model

As seen in the previous paragraph, the following elements for the model are crucial:

- The understanding of who was responsible for what was not clear.
- Goal setting and analysis through Architecture, Stakeholder requirements are crucial
- The business case should guide the value creation in the project
- All system levels have to be addressed
- Methodology as such is not important: bridging the gap between the methodology and what is needed to perform the full SDLC is.
- To understand project reports, this preliminary work and a working agreement improves reporting and questioning it.
- The Project Plan should be bases on that mutual understanding of which activities had to be carried out by whom.

These findings lead to an adjustment of the model:

- The mutual understanding of who is performing which tasks per system level in the preparation phase should be added. This mutual agreement is added as transitory “product” between the preliminary and delivery phase
- Contracting the delivery is an outcome of the preparation, not the start.
- The methodology should be followed as intended: this means that in the mutual agreement those artefacts have to be delivered to the products have to be ready to be able to comply with the methodology.
- All levels must be considered to create success. In all levels, the input for the ISO25010 requirements must be defined to ensure the functional requirements will be developed correctly. The quality measures to the functional requirements will originate from them. If neglected problems occur.

5 The adjusted and detailed model

5.1 Develop and Build: the model

The last question to answer is:

> *is it possible to construct a model to support project boards/clients in steering the project?*

With the sub-questions

- What do project boards/clients steer/what question should/could the client ask? And
- Can a model/questionnaire be derived for the client/project board?

In this chapter, a model is presented as a possible answer to these questions. The model is presented primarily for the PLAN or Initiation phase because the cases and the survey showed that steering a project well, starts with proper preparation of a project.

After this preparation, the client and the supplier have a full understanding of who is delivering what. This model supports and improves this process of this division of work and gaining understanding of the implications of that division of work.

The model is based on the conceptual model from chapter 4 and is combined with the artefacts that belong to the different system level

In this chapter, this model is developed further and justified by comparing it to the two case studies and 50 reviews performed by BIT (the Dutch ICT governmental review authority).

The construction led to new insights: Without asking questions, but using an imperative approach, model was more apt for all three phases in project management: Plan, Do and Check. The questions can be derived easily as is shown in section 1.4. Also, some underlying assumptions became clear. The most important one is that if a supplier is not able to clarify in a plain language, his actions and deliverables, the understanding between client and supplier cannot be established. The client does not need to fully understand the implications of a methodology, the supplier should tell them.

This research started with the actual question whether project boards need substantial knowledge of IT to steer a project and which deliverables they must understand.. By pursuing this research, it became clear that errors, omissions, misunderstandings in the preparation phase influenced the outcome in such a way that adjusting or intervening during the project was hardly possible. Not only the two case studies proved this, but the 38 cases of the survey also proved that the chance for success improved when especially the questions about the goal of the project were correctly answered
Even after a BIT-review, that recommended well-founded adjustments, some projects were not able to get on track again.

The construction of the model took four iterations based on the following requirements, based on the preceding chapters:

- The full system development cycle must be present in the model since, whatever method is being used, to all activities must be paid attention.
- all levels of the system must be addressed to produce the precise requirements for the IT system.
- the language used should be understandable by management at the corresponding levels without knowledge of IT (but with in-depth knowledge of their business).
- using the model must predict the outcome.

In the first iteration, the levels of system development (all aspects) and the ISO12207 activities were combined. All output of the activities, the deliverables, were attached to these combined activities. For all deliverables, questions were constructed in business language.

In the second iteration, the levels of contracting were added. By doing so, several deliverables showed to be dependent on each other. The contracting levels corresponded to the detailing of deliverables in system development.

The system layers (context, enterprise (the enterprise manifests itself through products and services, people and processes) also delivered the requirements for the ISO12207 activities (the IT level). The contracting levels turned out to resemble the strategic, tactical and operational management level at the three highest levels.
Combining these insights yielded a model in which the questions could be arranged in a new and structured way.

In the third iteration, the questions have been removed and replaced by actions. This description of actions fits better with the purpose of the model. By asking the matching questions, it is applicable in all three phases.

The description is constructed in business language for the following reasons :

- Defining what is precisely needed should be carried out carefully by the clients themselves.
- The definitions of what is needed should be unambiguous.
- Using IT terminology by not-IT-professionals can cause misunderstandings.
- It is the right of business owners that they get an answer in their vocabulary, that means, understandable without using technical terms.

The premise is that when a supplier or internal department is not able to answer these questions without using technical terms, the question of the business owner is not fully understood yet.
Above all: questions have to be answered at the right level. Providing details to answer the questions means that the problem is not clear enough. The client then has to combine the answers and fill in the gaps, which promotes miscommunication.

In the fourth iteration, the model was analysed by comparing the questions with the two case studies and the 50 reviews of BIT. These projects are by nature solely Public projects, information on private projects is hardly available.
This analysis caused a slight adjustment caused by the findings of the case studies and the analysis of the BIT-reviews.

At the Purpose and ICT level, questions have been added on

- As-is-to-be and migration planning at the purpose level.
- Implementation planning at the usage level.
- Emphasis on the ISO25010 requirements, especially security, privacy, maintainability and performance.

Implicitly these questions are handled by executing the architecture discipline. Even when delivering an architecture is almost obligatory, the problems that occur indicate that the architecture is not always good enough or not executed at all.

After the iterations, the levels of contracts were renamed into "levels of responsibility". The renaming resembles the situation of transferring responsibility to an internal (or hired) program manager, project leader or department and the management levels better. This renaming also resembles the levels of management (strategic, tactic, operational).

Another important finding from the BIT-reviews is that sufficient and skilled staff is essential. By having it addressed at the expertise level, it shows that the skill set needed is extensive. This list of skills is not meant as exhaustive. Each organisation uses its particular nomenclature for job roles. The essential message of this expertise level is that at each level specific expertise is necessary to provide the precise requirements:
IT specialists are no legal experts, nor marketing specialists whatsoever: they have to build a state-of-the-art solution fitting to these requirements and adjust their process to the specific situation of the organisation.
The next section describes the final design of the model. The construction of the model took some iterations. In this section, the iterative process to deliver the model is described. To improve the readability of this section, the intermediate results can be found in Appendix 4.

In the third section, the model is justified twice, first, by showing the results of answering the questions for the two case studies, secondly, by comparing the results of the reviews of the BIT-bureau with the questions of the model.

In the fourth section, the usage of the model is explained.

The journey to deliver this model does not end here. It is the base for a new concept to deliver courses to clients who have to steer projects.

5.2 Towards a mutual agreement, final design

5.2.1 Introduction

The mutual agreement is a tool to improve the quality of the arrangements in a contract. A model has been produced to draw up this agreement, divided into a 5*5 diagram. This diagram is based on the system and contract levels. It takes into account all activities that should produce outputs to deliver a successful project.

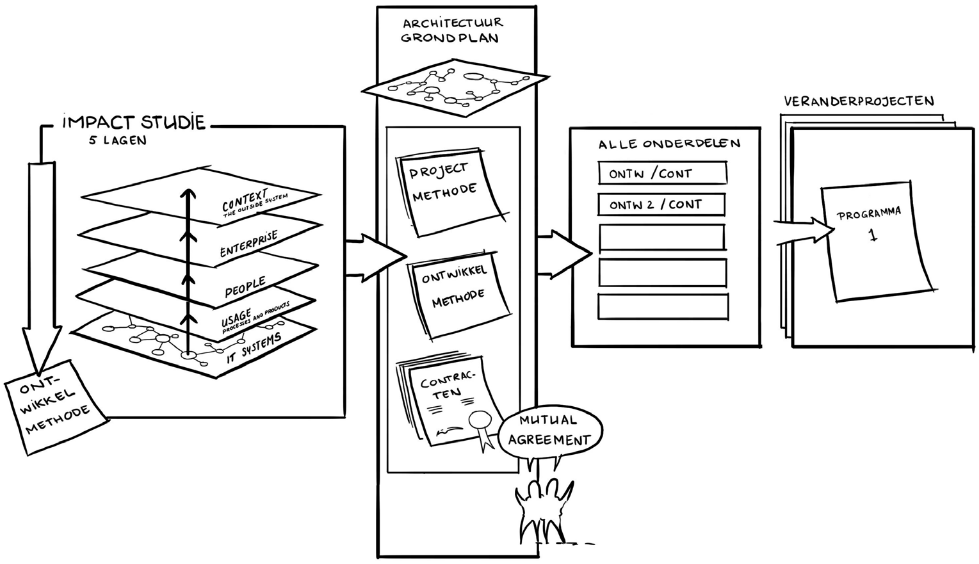


figure 8

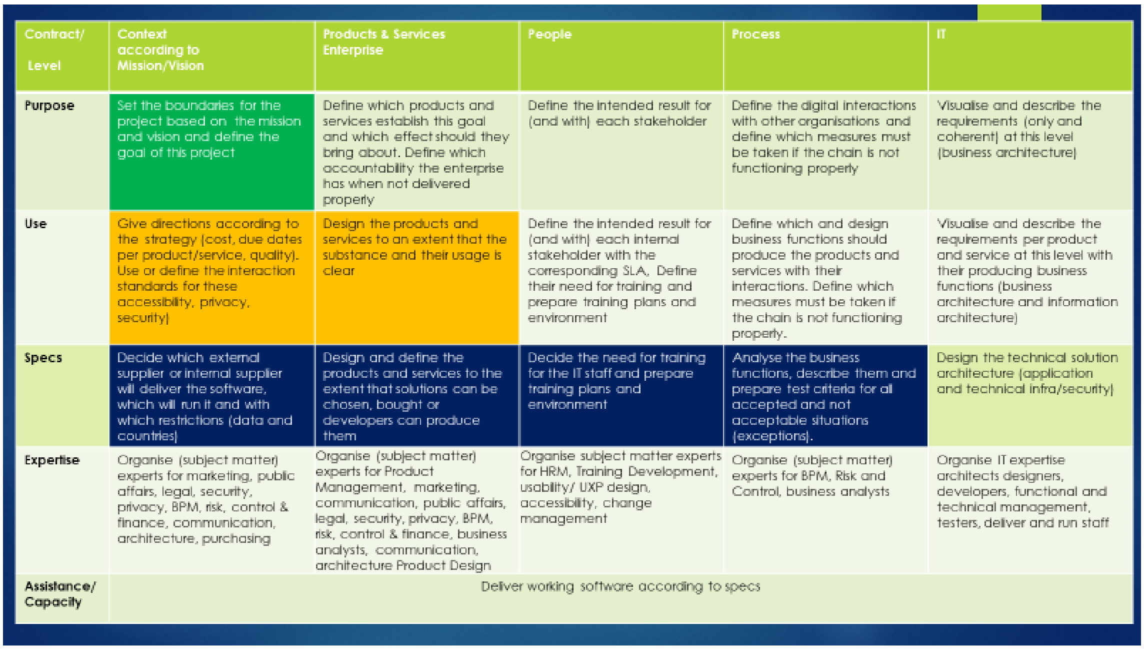
The chosen methodology decides the order of the execution of activities. The methodology is the third axe in the cube.

A mutual agreement implies that both parties should take their responsibility. The client has to deliver the answers on time. The supplier has to ask the right questions.

The client has to decide which responsibility they want to hand over to the supplier, and the supplier has to decide if he can take this responsibility fully. If not, it has to be discussed beforehand and recorded what the involvement of the client should be, to which extent and when it should be delivered.

5.2.2 The contract boxes and the mutual agreement

The main layout starts with the contract levels. Of course, different contracts at different levels could be closed if some parts need another approach. To decide what is best in a specific case, it might require some elaboration of the other topics.



Contract/ Level	Context according to Mission/Vision	Products & Services Enterprise	People	Process	IT
Purpose	Set the boundaries for the project based on the mission and vision and define the goal of this project	Define which products and services establish this goal and which effect should they bring about. Define which accountability the enterprise has when not delivered properly	Define the intended result for (and with) each stakeholder	Define the digital interactions with other organisations and define which measures must be taken if the chain is not functioning properly	Visualise and describe the requirements (only and coherent) at this level (business architecture)
Use	Give directions according to the strategy (cost, due dates per product/service, quality). Use or define the interaction standards for these accessibility, privacy, security)	Design the products and services to an extent that the substance and their usage is clear	Define the intended result for (and with) each internal stakeholder with the corresponding SLA. Define their need for training and prepare training plans and environment	Define which and design business functions should produce the products and services with their interactions. Define which measures must be taken if the chain is not functioning properly.	Visualise and describe the requirements per product and service at this level with their producing business functions (business architecture and information architecture)
Specs	Decide which external supplier or internal supplier will deliver the software, which will run it and with which restrictions (data and countries)	Design and define the products and services to the extent that solutions can be chosen, bought or developers can produce them	Decide the need for training for the IT staff and prepare training plans and environment	Analyse the business functions, describe them and prepare test criteria for all accepted and not acceptable situations (exceptions).	Design the technical solution architecture (application and technical infra/security)
Expertise	Organise (subject matter) experts for marketing, public affairs, legal, security, privacy, BPM, risk, control & finance, communication, architecture, purchasing	Organise (subject matter) experts for Product Management, marketing, communication, public affairs, legal, security, privacy, BPM, risk, control & finance, business analysts, communication, architecture Product Design	Organise subject matter experts for HRM, Training Development, usability/ UXP design, accessibility, change management	Organise (subject matter) experts for BPM, Risk and Control, business analysts	Organise IT expertise architects designers, developers, functional and technical management, testers, deliver and run staff
Assistance/ Capacity	Deliver working software according to specs				

Figure 9

Before discussing the 5*5 square model, it should be emphasised that this conceptual framework is constructed to decide which activities will be carried out by whom. It does not show a prescriptive path or order. The chosen methodology decides the path. However, some methodologies do not cover certain activities, as is shown before. Moreover, as is seen, the misunderstanding of who is responsible for completing these activities causes a weak starting position for the project.

The model shows two axes: the horizontal system-level axe and the vertical responsibility axe. The horizontal axe shows the five levels of the whole system. The responsibility axe shows what has to be accomplished to take responsibility from that level.

The framework shows most conditions under which the project can be executed with a higher chance of success.

The division of work alone is not enough: based on this division, the client and supplier have to agree upon a transformation plan. This transformation plan is based on the difference in architecture of the as- is and the to-be situation and the value to be delivered.

This model offers the possibility to use the framework in all sizes of projects and is thus independent of the size of the "project": a sprint, as well as a full-size project, fits. In all cases, small or big, it is crucial to understand whether all conditions are met. That all conditions are met can be proven by implementing (small) parts of the total solution that meet all conditions and can be scaled and clicked into the total solution. Dividing the total solution into those smaller or even tiny parts in the right order requires thorough planning and presupposes a well-defined architecture.

Special attention has been paid to ensure that the formulation is in non-IT language. The underlying hypothesis is that the supplier answers accordingly. The premise is that, when a supplier is not able to do so, the understanding of the client’s question is insufficient.

The model should be read as follows:

At the highest level, **purpose**, the five squares contain the questions about the purpose of the project at all levels: This is the realm of strategic management.

At the *contextual* level, the question is how the company manifests itself in the world, as the mission and vision of the company should explain. Whatever contract level is chosen, the question of how this project contributes to the mission and vision of the enterprise should be answered. The mission and vision determine the goal of the project. These are the leading principles for the project.

For the next four squares both have to decide who takes which responsibility:

At the *enterprise* level, the products and services are defined that effectuates this goal. Is the supplier able to take the responsibility to define the products and is the client willing to relinquish it?

The same applies at the *people* level: does the client dare to outsource the relationship building and negotiations with the stakeholders at this level to the supplier and are they able to take this responsibility?

To decide what measures must be taken if the digital interactions with other organisations are not functioning requires not only the definition of those interactions but also the understanding of the risk of reputational damage for the client. Based on the chance that this risk occurs and if when it occurs, the supplier can take this responsibility to solve it, the decision should be made whether to outsource this responsibility.

The last level, the IT column uses the requirements of the upper levels to transform it into an IT solution based on the ISO12207 order. The ISO25010 should be explicitly added or asked for if not mentioned yet in the terms of this level. This summary must be translated into understandable language. If the client and supplier agree upon the outcome, the next level can start.

This level is the most important level as we have seen to get thing right from the start: therefore a good summary of the starting points is essential.

If one or more questions result in the wish that the client himself performs the execution (whether assisted by a specific professional or not) than this level of sourcing cannot be chosen and at least the two orange boxes at the next level have to be carried out by the client.

When the client transfers all responsibility for the outcome to the supplier, they will realise the full business case within the boundaries of the mission and vision of the organisation. If for instance the organisation signed the *UN Guiding Principles on Business and Human Rights*, the supplier should guarantee that the same principles are applied and the client has to be informed (and probably make it possible to check) that these rules are followed. To define the mission and vision detailed enough for this purpose is thus essential.

Sometimes clients hand over the full responsibility to a program manager. In that case, they must have full freedom to perform their job, not impeded by internal agreements.

Transferring the responsibility at this level means defining the green square very carefully since this is the only possibility to steer. This defining is extra vital since mostly these types of contracts are long-term contracts and buying off these contracts can be rather expensive. For other projects ist is essential as well but the feedback loop is easier.

The business case for this level of contracting should be clear as well; some work of the levels below could be necessary to execute to set up this business case to understand if the contract is fruitfully for the client and supplier as well.

The same procedure is performed at the Usage level. Handing over the full responsibility to the supplier makes them accountable for the correct usage by all people that use the system. The conditions to perform this task must be derived from the answers to the questions on the upper layer and the first two boxes.

When the client did not choose for the purpose level *cont*ract, at this level, the contextual strategy has to be developed by the client and applied to the project. These should contain directions on cost, due dates per product/service or quality and provide the company's standards. that guide the usage of the system.

On the *enterprise* level, it means that the product and services design has to be provided at that extent that the requirements are unambiguous on what proper usage means. The client and the supplier have to decide who is responsible for the results per product and service internally (externally is already defined in the previous level).

For the people level the tactical management layer gets involved: they are responsible for the production of these products and services. It implies that, when taken over by supplier, the supplier decides how the results are obtained, which training is needed, not the client. KPI-setting per role could be part of it. Acceptance criteria have to be defined

On the *process* level, client and supplier decide who is responsible for the design of the business functions that will produce the products and services with their interactions and the fall-back-procedures.

The *IT* level again translates all requirements into IT requirements per product and service embedded in their business functions (business architecture and information architecture). ISO25010 requirements are added and flaws, omissions and contradictions are fed back to the stakeholders.

If for one of the light green boxes is answered that the client will be responsible for these actions, it means that this usage type of sourcing is not suitable for the client. It means that the client has to ensure all activities that belong to these boxes.

At the **Specs** level**,** all boxes have to be completed. The operational management is involved in getting the specifications right within the boundaries of strategic and tactical decisions. Specifying at this level gives the client more control over the execution of the project but requires timely delivery. Otherwise, it slows down the delivery of the project.

At *the contextual* level, it makes a difference for the specifications whether the supplier of the software is only delivering the software or is also running it, internally or externally. In both cases, additional requirements could be needed. Contracting could be due to formal procedures (like tenders, for instance),

At the *enterprise* level, the specifications of the products and services need a granularity to make it possible to make the right decision on the use of open source, standard applications, producing custom software or a combination of those. The conceptual application architecture must underpin these choices. When the client insists that they will be in charge of choosing the right mix, the specs contract is not fit for this client. Otherwise, the supplier can choose the best combination within the restrictions the client has given in the first box and the levels above (summarized in the architecture).

At this *people* level, the IT department (within the organisation of the client or the supplier) must be able to accept and manage the solution after completion of the project.

At the *process* level, the analysis of the business functions using or producing new or altered products and services must lead to the preparation of the test criteria for all situations, also the technical failures. .When the supplier delivers the solution according to the specifications, the client still has to provide the acceptance criteria.

If the client wants to use his influence on all the boxes (thus also the last) a contract with delivery according to specifications is not suitable.

The two other levels are different since the execution of those levels takes place when the decision is taken to start the project. Hiring **Expertise** means that the experts should have individual freedom of and influence in realising the best solution within the boundaries of the specifications. The experts can be working in all levels above. However, this does not transfer the responsibility to the expert. They have to provide quality advice; if not followed up, they have to advise on the consequences and if still not followed up to withdraw themselves from the assignment. In the boxes are all types of experts mentioned that are minimally required to accomplish the tasks mentioned in the upper levels.

Relevant to transfer knowledge to clients' organization is to decide on people level how this knowledge transfer is assured.

The **Assistance or Capacity level** means that the steering is the responsibility of the client and that the assisting professional acts according to his professional standards.

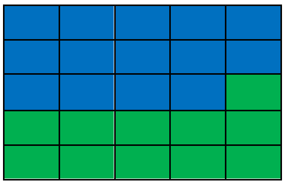
When hiring a project manager, the contract with them, is usually at the expert level. The total responsibility for a project can only be handed over when they have the full freedom to act according to the agreed contractual level. It also means that they are solely accountable if the project does not deliver the required results. This accountability has to be arranged officially. At this level, the ratio of internal and external professionals has to be considered. The e-CF can guide in choosing the right capabilities. Careful selection is crucial, as well.

When hiring a project manager, the contract with them, is usually at the expert level. The total responsibility for a project can only be handed over when they have the full freedom to act according to the agreed contractual level. It also means that they are solely accountable if the project does not deliver the required results. This accountability has to be arranged officially.

5.2.3 Adding the methodology

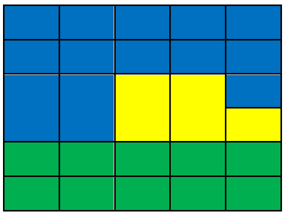
The methodology is important for the division-of-work decisions during the Plan phase but is essential during the Do phase:

When using scrum, the two lower layers are indeed covered. Professional teams will deliver the working software. Composing the team is an essential activity in Scrum. The business is connected through the product owner and the demo’s. The requirements have to be delivered by the product owner (the specs level). Together with the users, these requirements are translated into user stories, which together should sum up to business functions. The product owner determines which user story has priority. The pace in which these requirements are delivered, amongst other criteria, the velocity of the teams. Waist can occur when the requirements are not ready on time.



The product owner always acts on behalf of or is the client. The product owner is responsible at Purpose or Usage level and cannot transfer this responsibility. It is the task of the product owners to guarantee that the user stories sum up to the total. Also is their responsibility that the users or customers accept the result. (blue: product owner, green: teams)

Scrum teams often work in other frameworks, like SAFe, Spotify or Less. These frameworks give guidance to the client how they can organise the first three levels. These frameworks do not describe any methodology or give technical guidance on deliverables: it defines only outcome like epics, features or architecture.



When using RUP, the inception phase delivers a Vision document in which the business functions are analysed. It also delivers a software development plan based on architecture (blue). In the Elaboration phase, the specifications are completed, and the software architecture is proven (yellow). When the yellow and green squares are outsourced, it implies a specs contract; otherwise, it can only be sourced by hiring expertise or capacity.

When using Waterfall, all steps can be handled, and all contract forms are possible.

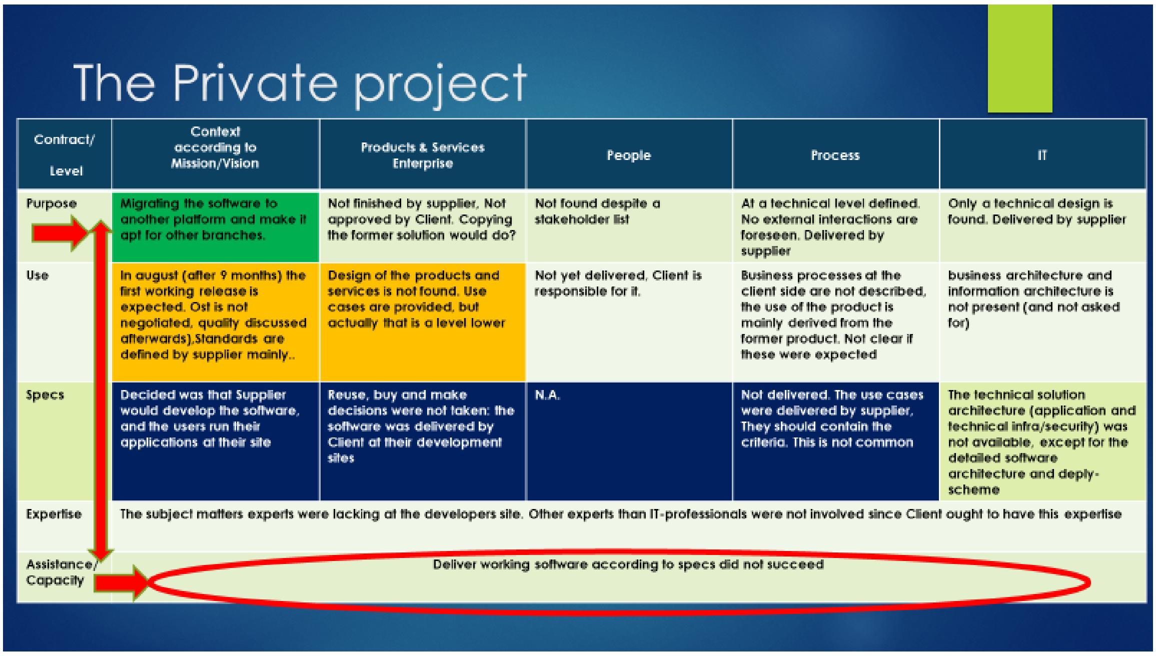
5.3 Two-fold justification and evaluation

5.3.1 Applying this model to the two case studies

When completing the boxes with the status of the projects, it can easily be seen that both projects were in danger.

In both cases the purpose level contained flaws and omissions. Therefore the requirements could not be complete, correct and sufficient. This research is not about showing how this could be rectified during the project. However, it certainly would have helped when all impediments had been openly discussed through analysing at the Specs and Use level what was missing and which choices had to be made and discuss this transparently with the Client. This discussion only took place four months before going live.

In the **Private** case, Client got a team with experts (project leader, analysts, architects and a team of developers), however. it seemed that the Supplier took the responsibility in steering the project. The contract itself was a contract at the capacity level, not expertise. Which meant Client had to provide all the necessary input and to define all requirements, even specifying the needed expertise. He was convinced he did, by hiring the project manager, a business analyst, software architects and other professionals of Supplier. Even confusing was that Supplier did not want him to steer the development teams and at the same time appointed a delivery manager. The appointed project manager composed his team without interference of Client.



The Private project

Contract/ Level	Context according to Mission/Vision	Products & Services Enterprise	People	Process	IT
Purpose	Migrating the software to another platform and make it apt for other branches.	Not finished by supplier, Not approved by Client. Copying the former solution would do?	Not found despite a stakeholder list	At a technical level defined. No external interactions are foreseen. Delivered by supplier	Only a technical design is found. Delivered by supplier
Use	In august (after 9 months) the first working release is expected. Ost is not negotiated, quality discussed afterwards),Standards are defined by supplier mainly..	Design of the products and services is not found. Use cases are provided, but actually that is a level lower	Not yet delivered, Client is responsible for it.	Business processes at the client side are not described, the use of the product is mainly derived from the former product. Not clear if these were expected	business architecture and information architecture is not present (and not asked for)
Specs	Decided was that Supplier would develop the software, and the users run their applications at their site	Reuse, buy and make decisions were not taken: the software was delivered by Client at their development sites	N.A.	Not delivered. The use cases were delivered by supplier, They should contain the criteria. This is not common	The technical solution architecture (application and technical infra/security) was not available, except for the detailed software architecture and deply-scheme
Expertise	The subject matters experts were lacking at the developers site. Other experts than IT-professionals were not involved since Client ought to have this expertise				
Assistance/ Capacity	Deliver working software according to specs did not succeed				

Figure 10

At the contextual level (in the vertical direction) the goals and directions were rather clear, even if one could have some remarks on the real business case and the experience to run the new operation.

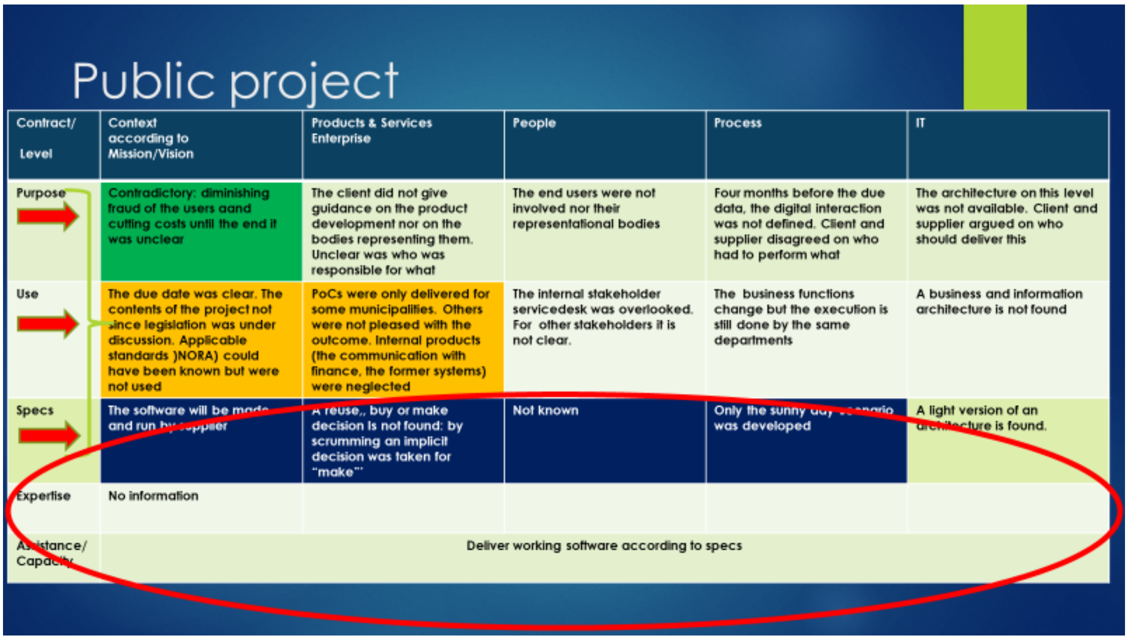
At all other columns of the purpose, use and specs problems occurred. The idea was to copy the old application to the new one. By refactoring the software, the products and services that would be delivered changed slightly and the software developers could not understand the functioning of the software. A difference in the cultural background might have contributed to the lacking knowledge of the branch.

Supplier tried to provide input documents mainly at the specs level. Client did not timely react on any of the documents. Especially the processes are underexposed. By providing the old application, the approach was technical instead of functional. One could say the supplier could have advised hiring other professionals, like UXP-designers and process designers. Subject matter expertise was not available, as well.

It is rather evident that the purpose, use and specs level are not detailed enough to get a successful product after nine months. It is puzzling that the situation was not clarified earlier. The project ended after four years, the spending of 2.3 million euro's and without working software (which was disputed by Supplier) and a huge dispute.

At the **Public** project, the outcome was the same, but the cause was different.

In this case, the purpose of the project was to prevent fraud. Parliament changed the law several times and until two weeks before the first due date. Later investigations showed that most users of the arrangements did not cause fraud, but some care organisations did deliver and underperform deliberately. The total legal system and the chain of information exchange were changed drastically. An architectural design of the new chain was not delivered and who was responsible for what was discussed until the due date. The delivery of that architecture was only completed in 2018! Designing processes and software architectures in such circumstances is hardly possible.



Public project

Contract/ Level	Context according to Mission/Vision	Products & Services Enterprise	People	Process	IT
Purpose	Contradictory: diminishing fraud of the users aand cutting costs until the end it was unclear	The client did not give guidance on the product development nor on the bodies representing them. Unclear was who was responsible for what	The end users were not involved nor their representational bodies	Four months before the due data, the digital interaction was not defined. Client and supplier disagreed on who had to perform what	The architecture on this level was not available. Client and supplier argued on who should deliver this
Use	The due date was clear. The contents of the project not since legislation was under discussion. Applicable standards)NORA) could have been known but were not used	PoCs were only delivered for some municipalities. Others were not pleased with the outcome. Internal products (the communication with finance, the former systems) were neglected	The internal stakeholder servicedesk was overlooked. For other stakeholders it is not clear.	The business functions change but the execution is still done by the same departments	A business and information architecture is not found
Specs	The software will be made and run by supplier	A reuse,, buy or make decision Is not found: by scrumming an implicit decision was taken for "make"'	Not known	Only the sunny day scenario was developed	A light version of an architecture is found.
Expertise	No information				
Assistance/ Capacity	Deliver working software according to specs				

Figure 11

It was even unclear who the real client was: was it the ministry under which the supplier resided or the ministry under which the law was changed? Other ministries could have taken the lead also. The discussions with stakeholders took place at all levels, but decision making was unclear and discussions unfinished.

Client claimed that using Scrum could solve these issues because only working parts would be delivered. Instead, they chose for "a sunny day scenario" which implies that all exceptions are not present.

By studying the BIT audits (see next section) that were pursued twice after this implementation, it became clear that the same issues still exist and that the project has to be delivered and housed elsewhere which is not the only solution to the problem of quarrelling stakeholders.

5.3.2 Applying this model to the BIT reviews

BIT, the Bureau of ICT reviews, acting on behalf of the Dutch government, reviewed in the past four years, 50 projects that trespassed the limit of five million euros to spend on IT.

These 50 projects have been numbered according to the list in the table below, and the conclusion of the reviewers have been translated into colours:

- Red: The project should stop
- Orange: The project needs firm intervention
- Yellow: The project needs some adjustments
- Blue: The project is unnecessary
- Green: The project is on track and probably will deliver results

Of course, it is not entirely clear in some situations if the reviewers or the client made the description on the purpose and context of the organisation and the project. Only a glimpse of the real situation can be seen, but it provides some valuable insights that cannot be found from private projects.

Where possible, the results at this moment were checked at the ICT dashboard of the government website 25). Some projects did not deliver, and of two of them is known that the advice was hardly followed (PGB and NVWA) They still are not functioning well.

- Only 5 out of 50 projects were considered as having a high success rate (and were successful).
- 12 projects would possibly succeed when they would take the advice seriously
- 18 projects were in severe danger and had to make substantial adjustments
- 16 projects were advised to stop and reconsider their purpose, of which four were advised not to start at all since they were unnecessary.

	context	P&S enterprise	people	process	IT
purpose	4, 7, 14, 25, 29, 35, 41, 42, 43, 44, 46, 48	1, 4, 7, 10, 21, 29, 31, 32, 35, 41, 43, 44, 48	1, 13, 29, 35	1, 4, 10, 16, 28, 31, 42, 43, 44, 46, 47, 48	4, 7, 10, 13, 15, 16, 23, 25, 41, 42, 44, 47, 48
use	1, 4, 7, 10, 14, 15, 21, 29, 31, 46	1, 4, 10, 15	1, 13	1, 4, 10, 16, 23, 25, 31, 42, 44	4, 10, 13, 14, 15, 16, 17, 23, 25, 41, 47, 48
specs	1, 16, 21, 25, 29, 31, 45, 46	1	7, 13, 21	1, 13, 15, 23, 25, 42	4, 10, 13, 15, 16, 21, 25, 41, 46, 47
expertise	13, 31				4, 7, 13, 31, 41, 45
capacity					

Table 9

When putting the projects in the boxes, it can easily be seen that the problems occur mainly at the highest level, and have their impact on the IT levels. BIT emphasises a lot on the implementation planning, and the work break down structure for that. This matches with the problems that arise by not defining the products and services well (and that is not IT). A good product description gives guidance to the implementation plan and provides an insight which products are essential to start with and which quality level is necessary to use the product or service correctly. The implementation planning is derived from the migration architecture which should be considered already from the purpose level.

It is somewhat surprising that in most cases, the processes are not or not sufficiently described to give guidance to the projects. This finding could lead to other research. It Is not surprising that (the lack of good) project management turned out to be an issue as well, especially the adequate reporting during the project. This research is not about project management but on how to provide tools from the substantive approach. However, they influence each other.

Leaving out the yellow and green projects shows that especially In the purpose and context level most problems occur which implies IT architectural problems. This promotes outside in thinking for projects.

Leaving out the yellow and green projects shows that especially In the purpose and context level most problems occur which leads to IT architectural problems. This promotes outside-in thinking for projects.

purpose	context	P&S enterprise	people	process	IT
purpose	4, 25, 29, 41, 42, 43, 44, 46, 48	4, 7, 10, 21, 29, 31, 41, 43, 44, 48	13, 29	4, 10, 16, 31, 42, 43, 44, 46, 47, 48	4, 7, 10, 13, 16, 25, 41, 42, 44, 47, 48
use	4, 7, 10, 14, 15, 21, 29, 31, 46	4, 10	13	4, 10, 16, 25, 31, 42, 44	4, 10, 13, 14, 16, 25, 41, 47, 48
specs	1, 16, 21, 25, 29, 31, 46	1	7, 13, 21	13, 25, 42	4, 10, 13, 16, 21, 25, 41, 46, 47
expertise	13, 31				4, 7, 13, 31, 41
capacity					

Table 10

The analysed projects in the list are all projects that have been reviewed by BIT form 2015 until now:

Nr	BIT- review	Phase	Review date	rating
1	Centralised Base Luchtverkeersleiding Nederland	Plan	8-5-2019	🟨
2	Kern Gezond	Do	30-4-2019	🟥
3	Processen, Informatie en ICT (PI&I)	Do	3-4-2019	🟥
4	European Rail Traffic Management System (ERTMS)	Plan	26-3-2019	🟧
5	Basisregistratie Ondergrond (BRO)	Do	23-1-2019	🟨
6	Gemeenschappelijke Centrale Meldkamer	Do	18-1-2019	🟥
7	Vernieuwd Praeventis	Plan	17-1-2019	⬛
8	Inleenadministratie en Quotumheffing	Do	10-12-2018	🟨
9	Verwijzingsportaal Bankgegevens	Check	10-12-2018	🟨
10	Maritiem Operatiecentrum (MOC) Kustwacht	Plan	29-11-2018	🟥
11	Bediening op Afstand, sluizen en bruggen in Friesland	Check	2-11-2018	🟨
12	Persoonsgebonden Budget 2.0	Do	15-10-2018	🟥
13	DigiInhuur	Plan	10-10-2018	🟥
14	JDS	Plan	12-9-2018	🟦
15	Transitie Werk.nl	Plan	23-8-2018	🟨
16	Rehosting AGS/DTV	Plan	10-7-2018	🟦
17	Programmaportfolio werkplekdiensten	Check	25-6-2018	🟨
18	ICT-werkplekdienst- verlening	Do	25-6-2018	🟧
19	iTEC-based Centre Automation System (iCAS)	Do	12-6-2018	🟨
20	Landelijk Meetnet Water 2	Do	31-5-2018	🟨
21	Grensverleggende IT (vervolgtoets)	Plan	30-4-2018	🟧
22	Eén uniforme betaalomgeving – WW-deel	Do	26-2-2018	🟩
23	Phoenix+	Plan	20-12-2017	🟨
24	Wet tegemoetkomingen loondomein	Do	7-11-2017	🟥
25	IV van Morgen	Plan	18-10-2017	🟧
26	Electronic Flight Strips	Do	17-10-2017	🟩
27	Digitaal Stelsel Omgevingswet	Do	4-9-2017	🟧
28	CivMil Voice Communication System	Plan	1-8-2017	🟩
29	Investeringsagenda Belastingdienst	Plan	13-6-2017	🟦
30	Operatie BRP (oBRP) en In Beheer Name BRP (IBN BRP)	Do	9-6-2017	🟥
31	Aanbesteding datacenter UWV	Plan	28-3-2017	🟧
32	DWR Next	Plan	21-3-2017	🟨
33	IT IGO KMar	Do	21-3-2017	🟧
34	Blik op NVWA 2017 (vervolgtoets)	Do	20-3-2017	🟧

Nr	BIT-review	Phase	Review date	rating
35	Lerarenregister en Registervoorportaal	Plan	15-3-2017	🟨
36	Implementatie Vernieuwing C2000	Do	22-11-2016	🟧
37	Harmonisatie Applicaties en Rijksvastgoed Processen (HARP)	Do	11-11-2016	🟧
38	Union Customs Code (UCC), Douanewetboek van de Unie (DWU), Multi-annual Strategie Plan (MASP) (Douane)	Do	3-11-2016	🟥
39	vAKWerk SVB	Do	18-10-2016	🟩
40	Een Uniforme Betaalomgeving UWV	Do	22-6-2016	🟧
41	Grensverleggende IT (GrIT)	Plan	31-5-2016	🟧
42	Directe Financiering Kinderopvang	Plan	20-5-2016	🟧
43	eID	Plan	13-5-2016	🟧
44	Zelfbediening Justitiabelen DJI	Plan	28-4-2016	🟧
45	Doorontwikkelen Basisregistraties Onderwijs	Plan	25-4-2016	🟩
46	Zaakgericht Werken DICTU	Plan	11-2-2016	🟧
47	Persoonsgebonden Budget SVB		26-1-2016	🟥
48	Blik op NVWA 2015 (pilot)	Plan	29-10-2015	🟥
49	Operatie BRP (Basisregistratie Personen)(pilot)	Do	18-9-2015	🟧
50	Omgevingsloket (pilot)	Do	17-9-2015	🟧

Table 11

5.3.3 Evaluation

The model gives a good indication of future problems:
In the two case studies, it is evident that at the highest system and responsibility level, the division of activities was unclear, or the activities itself not ready or not discussed. In both cases, this led to problems. Programming before completing these activities worsened the situation, and it certainly did not deliver enough value.

In both cases, the work should have been paused until clarity was given. In public projects, this is sometimes hardly possible because of the higher purpose (a new law, political discussions and goals). Even then, the model could help by identifying those products and services that are not impacted by those decisions and start with them instead of all products and services. In public services, one cannot apply the 80/20 rule easily: all civilians have to be treated equally in the same circumstances.

The BIT-reviews show the same results: especially the outer levels are not adequately executed, and this impacts the underlying levels.
The total amount of well-run projects (five) is too low to conclude, but it could be a good indication that the model works since it seems that the boxes at the outer levels are correctly completed.
Project management is not the topic of this research; however, the reporting on the projects is often inadequate. This model could give guidance to the reporting.

The indicative actions work as well in the Plan, Do and Check phase. When BIT started, many projects were already running. Nowadays, projects have to request a review at the start of a project. Then the effect can be measured. Because the projects were in different phases the model could be applied to all three and occurred to give roughly the same outcome: at the upper and outer levels, the activities to obtain a good architecture and sufficient requirements to complete the project were lacking.

In this section, the last question is answered: *Can a model/questionnaire be derived for the client/project board?.*
The answer is a humble "yes". It turned out to become less a questionnaire than an action list, which fits better with the term "levels of responsibility".

5.4 The model in use

The model can be used in the three main processes of project management:

- Plan: to obtain a transformation planning with the division of work.
- Do: running the project by checking the completeness
- Check: to decide if all agreements in the project are met, the model can be used again.

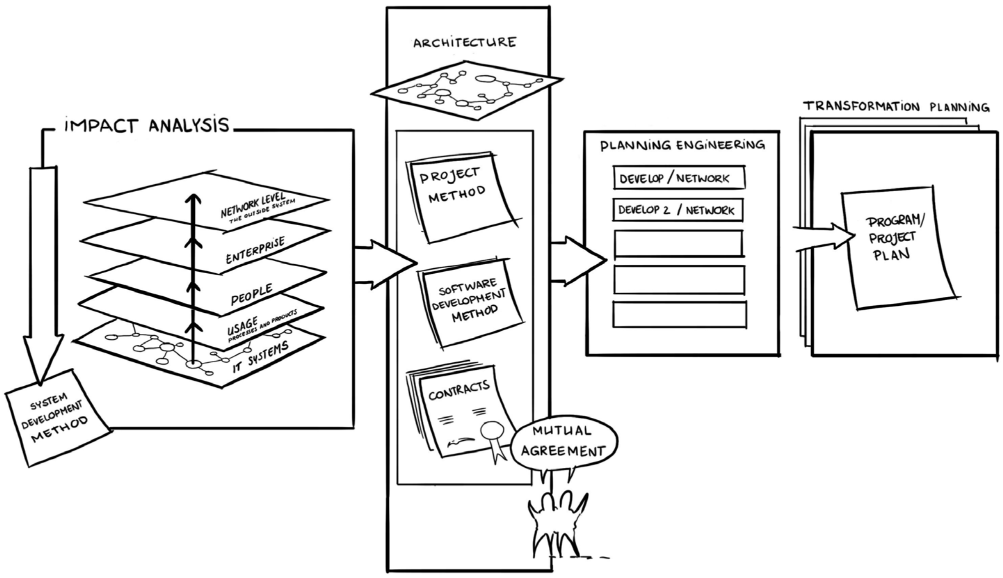


Figure 12

The Plan phase is defined as the phase in which the transformation planning will be delivered to realise the goal of the business owner.

Adding the methodologies to them gives guidance to use the model
When Scrum is used, the agreements are implicitly arranged by the role definitions.
Scrum presupposes capable teams that can prioritise their work based on the highest value when delivered. When transferring the work to the team, all previous levels have to be completed.

The business owner and product owners handle the Plan phase. They provide the necessary requirements at a pace such that the teams do not encounter any impediments. The requirements are delivered as features, and the teams decompose them into user stories which are approved by the product owner n behalf of the stakeholders and the business owner.

In some organisations, who steer multiple scrum teams, models are used to organise this work. For instance, SAFe is a well-known model (website 26). The highest level in full Safe (portfolio) is the purpose level, the next provides the same deliverables as the Usage level, and the program level can be compared to the specs level. If these models are not used, the organisation has to figure out how the requirements are delivered.

The Plan, Do and Check phase are overlapping like roof tiles, but each team is delivering at the same pace, delivering a working product every two or three weeks.

The contracting of work between the business owner and product owner on one side and the development teams on the other side can be done but requires close cooperation to take away impediments timely.

When using RUP, the division of work can be negotiated at any phase. The inception phase delivers at the purpose and usage level, the elaboration phase at the specification level and the construction phase at the expertise and capacity level.

RUP presupposes that the purpose level is ready, but this level is always checked and taken into account in the Inception phase. All contracts are possible: Purpose means that the Supplier coordinates and executes the Inception layer. Usage means that the supplier is responsible for the elaboration phase the construction and the transition. The Specs level implies that the supplier receives the specs from the elaboration and is responsible for the construction phase and the system test and functional acceptance test.

When using Waterfall, the work can be split anywhere. Waterfall is often used when specifications are to be constructed for many suppliers that need the same specifications.

In all cases, it is wise to start with an impact study. In this impact study, the 5*5 squares are completed to the extent that the client (or their project board) understand the impact of the project, can choose which transfer of responsibility they want and can steer the change with taking all aspects into account to deliver a successful change.

This impact study might result in one or more projects plans, or, using Scrum, in one or more epics or features using the most suitable methods and contract forms and ultimate date to deliver. The impact study is combined with enterprise architecture, which provides architectural migration paths by nature. This architecture must lead to coherent but well-distinguished parts that can be delivered within a maximum length of 9 months (Verhoef, 2002). When for each part, a specific methodology and project method is chosen the appointments are clear before the start (for instance: implementing a package often needs another approach than the delivery of custom made software).

In many cases, a tender has to be part of the start of projects. When deciding at which level the responsibility would have to be taken over by the supplier, a business case is needed to understand if the supplier can deliver enough value at the proposed level and to establish the boundaries in which the supplier acts.

By discussing each cube, the client can

- Establish the differences between the as-is and to-be situation
- Understand what has to be accomplished to reach the to-be situation
- Decide if they want to transfer the responsibility to accomplish the to be.

This agreement can be used to start the tender or first the RFI.

When a partner is chosen (externally or internally), this 5*5-document is used to specify the responsibilities and the outcomes per square further to an extent the client understands what is delivered, and what is not.
In this way, the consequences of not delivering substance by either party are immediately clear.

During the Do-phase, the completed 5*5 model and the mutual agreement are the backbones of the projects. As was seen in the case studies as well as in many BIT reviews is that reporting does often not show the real situation and is not eliciting the critical issues. These documents could be leading to risk management and quality management, starting from the highest level. Reporting on the 5*5-squares with a completion rate could be useful.

Before a project goes live, the final check is not only the user and production acceptance test but also the acceptance on the strategic level, the discharge of the project. For this, the model can serve to judge if the outcome is delivered as intended.

6 Results, discussion and recommendations for further research

6.1 Results

In this thesis the following research questions have been examined and answered:

- *Which deliverables or documentation (artefacts) of systems development projects are crucial for clients to steer upon, to get a higher chance of a good end result of a project?*
- *Also, if these can be found, is it possible to construct a model that could support Project boards in steering the project?*

The results are discussed by the hypothesis with which the research started.

Hypothesis 1: In the case of customised development, ISO25010 norms are part of the requirements.

When starting this research, the expectation was that the ISO25010-requirements (the quality aspects of software) were often overlooked and that this neglection mainly caused problems.

Executing this research showed that in many cases the conditions to get all requirements right (including the ISO25010) were not met.

The definition of the use of the word system in system development theories resembled almost language philosophy or general linguistics, but it showed the discrepancy in the use of the term "system". It even could be a root cause for the misunderstanding in what will be delivered.

This finding led to the first element of the model to be constructed: the system and its levels.

In the systems theory systems are consisting of levels and systems are interacting with the outside world. An information system is one of the layers and not the only layer. The requirements are gathered from all other layers. This implicitly requires the delivery of a software system small enough to deliver a working system at all levels and thus requires a thorough understanding of the interactions of the software and the levels (this is called (enterprise architecture).

For this model the whole system is subdivided in five layers:

- Contextual (where the organisation is interaction with the outside world)
- Enterprise (the products and services that define an organisation)
- People (people belong to enterprises and use processes and systems and are a factor in change)
- Processes (the formal or informal way of working to which an IT system must comply)
- IT (Application, infrastructure and maintenance)

Hypothesis 2: Methodology matters and relations between methodology, artefacts and activities are essential.

In both case studies the methodology was not executed according the methodology and the Clients did not notice. External quality audits at each step could have helped, but the problems occurred at an early stage.

The survey showed that many projects were delivered with the same method (Scrum) but had different results.

What turned out to be important is, **that depending on the chosen methodology the project initiation should cover the supplementing of the missing processes according ISO 12207**. This does not imply that all work has to be done first. But based on a good architecture the missing elements could be delivered during the project. Which elements, processes and artefacts have to be delivered has to be clear from the start. Also has to be decided who delivers what. This is the input for the planning and the project plan, without it, the planning is not controllable.

Therefore the conclusion is that the specific methodology is not a critical element. However acting according the methodology and checking what the difference is between the ISO12207 processes and the method is important to get a full view on the project plan.

This is supported by

Hypothesis 3: Quality management is present and positioned outside the program/project (in order for the business owner to get a clear view of the results).

The hypotheses above are related to the following hypotheses

Hypothesis 4: Successful programs/projects have, as a driver for the business case a clear goal that can be described by the business owner

Hypothesis 5: The constituent elements of the business case are known

Both case studies had problems with their goal setting: The private case had a clear goal, but veered of track (delivering a solution for the first user after 9 months). The public case had a goal set by parliament which changed during execution and appeared not to be he real problem (fraud by the care users).

The survey showed a convincing result that having a clear goal and being able to express it in one sentence led to working software.

An unclear business plan in the private case and the absence of an implication study in the public case are related. This is the realm of strategic management.

Hypothesis 6: To deliver the business case processes are known/developed before programming and alternatives are considered

Considering alternatives show better results. But to draw conclusions from this is more like an educating guess: considering alternatives is only possible If the business case is understood and a form of mastery based on architecture of the solution-to-be is reached.

The same is the case for processes: most respondents of the survey claim the processes are in place. The outcome of the projects seems not related. In the case studies however the processes were not developed. Whether that caused the problems or programming before all the necessary processes and artefacts were delivered is not clear. Definitely programming before knowing how to handle the project as a whole, just to gain time, is a contraporductive idea.

Hypothesis 7: Each type of contract needs its specific preparation up front.

In this research we also defined 5 levels of contracting.

1. Purpose (running the business).
2. Usage (turn key).
3. Conform specifications (with time/money constraint).
4. Expertise
5. Capacity

These levels are related to the system levels. The purpose level is taking over the whole operation and run according the strategic goals of the organisation in their environment. The Usage level is related to the products and services, the enterprise level that are delivered turn-key. The specification level requires all requirements to be delivered just in time.

The other levels are necessary when running the three different phases of the project.

It is clearly seen by the literature study that contracts deliver different results. And that adding these contracts to a specific methodology means that the substance of the excursion phase is quite different.

The conclusion therefore is that contract (and its levels) is a main element in the model

In the case studies the confusion on who was responsible for what mainly caused the problems. **Therefore a "mutual agreement" is introduced to decide on who is performing which activity.**

Hypothesis 8: Projects are not delivering results because clients lack knowledge about IT.

Actually it not the knowledge of IT that is lacking but the knowledge of the combination of contract, methodology and the expectation that the system will be running as a whole. The understanding of the word system and the different applications of the word in the different methods gives room for misunderstandings. **Therefore the model contains the constituent elements of a system.**

6.2 Discussion on the model

The results as discussed above, led to a model that might support clients and project boards by asking questions (and demanding answers that are understandable) for every cube in the 5 system layers * 5 steering or contract layers and the methodology. We call this model the Qube

In the initiating phase this must lead to a division of roles in completing the cubes before and during the project. This leads to a “mutual agreement” which is the base for the Project Plan.

The same Qube is used to check if all the cubes are completed.
The best way to complete the cubes is to realise relative small system parts that contain all cubes, since some cubes can be difficult because of the people aspect, others because of the IT aspect and other because the interfaces at contextual level take tome to validate. A good enterprise architecture with a accompanying migration architecture helps to divide the system in executable, performing units.

The model does not seem rocket science. However, many projects fail to succeed as intended or have serious problems. They did not answer these questions right . The analysis of the 50 project-reviews shows this quite clear.
Quality management to ensure that the answers are based on real progress, commissioned by the client, outside the project is essential.

The main research questions,

> *Which deliverables or content (artefacts) of systems development projects are crucial for clients to steer upon, to get a higher chance of a good end result of a project?.*
>
> *And if these can be found, is it possible to construct a model to support project boards/clients in steering the project?*

are answered in this thesis in a different way than expected:

Concerning the deliverables

Conducting the preparation phase right, before producing software seemed most important. Clients that act accordingly have a higher chance on a successful project.

In this phase the work needs to be done to that level that well-informed decisions can be made on the level of contracting, the appropriate methodology and the supposed business case or value.

The deliverables as such are not important: answering the questions is. This is a better phrase than agile principle : working software over comprehensive documentation: this leads to a shortage of deliverables and lots of discussion instead of analysis. It takes professional conduct to deliver those answers in transparent deliverables that makes quality assurance possible even when the client is not able to evaluate these deliverables properly. The reviews conducted by BIT show that Project boards, in which directors and other busy business managers usually take place, usually do not obtain full IT-knowledge to understand what deliverables are necessary.

The questions are based on the purpose of the standard deliverables and can be used to steer the quality assurance. They must be able to prove that the deliverables contribute to the answers without ambiguity.

Now almost all steering committees choose Scrum as methodology, it is rather important that is understood that the system realisation takes only place at the expertise and capacity level, actually at best effort. All the processes and deliverables of the other layers have to be delivered by the organisation itself by the business owner and for them, the product owners. Therefore the business has to be part of the development teams on a daily basis. The question is if that is really the case.

This is especially important since many organisations choose to hire a program manager only. Especially then it is important which responsibility is assigned to the program manager. If they have to act according the project board they actually act on a capacity level and the client is totally responsible for the outcome.

In the case studies we saw that suppliers did not act according to professional standards. At least the supplier should know what is needed to develop a system properly according to his own chosen methodology.

Concerning the model:

To assist in getting the right deliverables and to get the deliverables right to deliver working software but above all, a well-functioning system, the Qube-model is presented.

The model can be used at each stage but is meant especially for the first stage to construct the “mutual agreement”.

This mutual agreement is the base for the Project Plan, milestones alone are not enough. The progress on the project must be controllable. In the cases and from the BIT-reviews it is seen that the progress on projects is not well perceived. In the reviews many projects are perceived as successful although the indicators show otherwise.

This only shows the necessity of a better insight in projects and getting grip on the progress.

Therefore the model (and the mutual agreement) can be used as well to ask questions about that progress during the execution.

And at last, before closing down the project it can be used to assess the activities that should be completed or can be completed while in use.

The model is explicitly written in language for business owners. The answers should be given in the same language, underpinned by the deliverables that lead to the questions.

Each contract level resembles a management layer: purpose belongs to the strategic leve, usage to the tactical and the specs to the operational level. The other levels are the levels in the project and organisation that cooperate to get the right outcome: the teams and the people who are delivering the system. The questions on management level have to be answered by the management levels in order to understand the change and guide their staff through the change.

The model is already in use. It takes more than one swallow to make a summer. The intention is to use and improve it in trainings for project boards.

6.3 Recommendations for further research and in general

This research touches many points which requires more study, concerning

1. The professional and ethical standards of the supplier,
2. The tendering processes in relation to this model,
3. The difference between Lean/Agile methods in relation to the EFQM/Rhineland model,
4. The “governors’ bias”
5. Process Design, why not?
6. Learning from successful projects.

We touch them here briefly:

The professional and ethical standards of the supplier
In the private project the supplier didn’t act according to his own proposed systems development method. If he had, he would have stopped the project. By making the client totally responsible (which he actually was according the contract, but not in the way the supplier acted in the first months he avoided his own responsibility. Invoicing 2.3 Million euros without delivering working software is rather exceptional. The project was stopped when the client couldn’t pay anymore.

The private sector should establish an independent authority as well to judge the quality, to be provided by the sector itself. Other professional communities do have those instances, and to be a grown-up sector, this should be the case for the IT sector as well. BIT is acting only for the public sector and only for those projects in which IT spending exceeds 5 M€.

The tendering processes in relation to this model
Tendering processes in the public sector have been introduced to create a level playing field. The question is, at what level of detail the tender is written and which requirements are present when the tender is published. A study on the outcome of projects in relation to the quality of their requirements could provide insight in how to produce better tenders. Tendering often acts as if the contract level is known. But is the necessary work is not done, the spending will exceed the tender value, as is often the case according to the BIT-reviews.

The difference between Lean/Agile methods in relation to EFQM/Rhineland model
Lean/Agile methods stem from the Japanese methodology that was introduced by Toyota (the Toyota Production System). At the same time Agile methods were developed in the US. As an answer on the fast and growing Japanese presence on the European Market, with quality products at lower prices, the EU developed a system based on the European values (Rhineland model) but also based on the same Deming model that was used at Toyota. This was called the EFQM model (European Foundation for Quality Management). This model takes into account amongst others Society and Employee satisfaction. The difference in mindset causes differences in outcome of Scrum in the two societies. The way users get involved is interesting. Therefore it could be helpful to find out, which differences between the models have influence on the outcome of Scrum and how we can adapt the methods to the European model (and values).

The “governors’ bias”
In the survey and the case studies, governors have a bias towards reporting a more positive outcome than reality shows. It’s interesting to find out if this bias supports projects to focus on the right actions, is neutral or has a negative effect.

Process design, why not?

From the Bit-reviews and the cases we learned that many project lack a process design. This has a few implications like: building applications for an unoptimized situation and using technology that fits in the old situation instead of innovating with new possible techniques. The question is, why is this neglected so much?

Learning from successful projects
It could be helpful to understand what successful projects have in common. We focus always on the failed ones and assume that failure causes we found in their analysis, would have helped if only properly addressed. Analysing such factors in successful projects could help to show if that is the case. Actually it was the intention to learn from successful projects but the amount was rather low (5 out of 50 in the BIT-reviews and 6 out of 38 in the survey) which is not enough. Also, those projects were very different from nature.

From the above I am personally interested in studying the professional and ethical question.

LITERATURE

Websites/newspaper articles

1. https://www.rijksoverheid.nl/documenten/kamerstukken/2017/10/07/kamerbrief-arbitrageprocedure-svb
2. https://www.rijksoverheid.nl/documenten/kamerstukken/2017/11/09/kamerbrief-arbitrageprocedure-svb
3. Hoge Raad, 11 april 1986
   http://www.itenrecht.nl/www.delexbackoffice.nl/uploads/file/IT%20en%20Recht/HR%2011%20april%201986%20(RBC-Brinkers).pdf
4. Overheid: aanbestedingen
   a. Arbit 2016
      https://www.pianoo.nl/sites/default/files/documents/documents/arbit2016.pdf
   b. ARBIT, https://www.escrowalliance.nl/escrow-regelingen/arbit-escrowregeling/algemene-rijksvoorwaarden-bij-itovereenkomsten-arbit/
   c. Overheid pianoo: aanbestedingswet 2012
      http://wetten.overheid.nl/BWBR0032203/2016-07-01
5. Elias/kamer
   a. Elias, T. Parlementair onderzoek naar ICT-projecten bij de overheid, Tweede kamer, vergaderjaar 2014-2015, nr 33326-5
   b. https://www.tweedekamer.nl/sites/default/files/field_uploads/33326-5-Eindrapport_tcm181-239826.pdf
6. Bureau Gateway (2014)
   a. het gateway-proces in 5 stappen
      http://www.bureaugateway.nl/userfiles/file/Gateway%20Review%20-%20Het%20Gateway%20Reviewproces%20in%20vijf%20stappen%20(januari%202014).pdf#zoom=100
   b. Bureau gateway (2014) Producten en diensten catalogus
      http://www.bureaugateway.nl/userfiles/file/Gateway%20Review%20-%20Producten-%20&%20dienstencatalogus%20(juli%202014).pdf#zoom=100

7. https://www.surf.nl/binaries/content/assets/surf/nl/kennisbank/2015/201501---model-beleid-verwerking-persoonsgegevens.pdf
8. https://autoriteitpersoonsgegevens.nl/sites/default/files/atoms/files/verordening_2016_-_679_definitief.pdf)
9. https://www.computable.nl/artikel/nieuws/erp/6311855/250449/een-op-de-vijf-retailers-ziet-ict-project-mislukken.html (1 maart 2018)
10. http://www.fujitsu.com/nl/about/resources/news/press-releases/2018/emeai-20180226-fujitsu-survey-reveals-failed-digitalization.html
11. https://www.bureauicttoetsing.nl/afgeronde-adviezen/r/realisatie-it-ondersteuning-voor-dienstveriening-pgb
12. https://www.nrc.nl/nieuws/2017/09/29/nieuwe-website-voor-pgbs-vertraagd-door-stelselmatige-tegenwerking-13254764-a1575464
13. https://vng.nl/onderwerpenindex/sociaal-domein/persoonsgebonden-budget-pgb/nieuws/invoering-nieuw-pgb-systeem-vindt-stapsgewijs-plaats-in-2018
14. https://nos.nl/nieuwsuur/artikel/2014539-onzekerheid-rond-uitbetaling-van-250-000-zorgverleners.html
15. https://www.volkskrant.nl/nieuws-achtergrond/topvrouw-sociale-verzekeringsbank-weggestuurd-na-pgb-falen~b76e0ee2/

16. http://www.opengroup.org/it4it
17. https://www.vanharen.net/Player/eKnowledge/een_nieuwe_rol_voor_het_informatiemanagement.pdf
18. https://www.computable.nl/artikel/praktijkcases/management/1300202/1508217/aveacutero-legt-verantwoordelijkheid-ict-bij-afdelingsmanagers.html
19. http://www.cs.vu.nl/~x/ipm/ipm.pdf
20. http://www.house-of-control.nl/destep-model.html
21. https://www.scaledagileframework.com/glossary/#V
22. https://www.opengroup.org/it4it
23. https://www.agconnect.nl/blog/het-amsterdams-negenvlak
24. https://www.youtube.com/channel/UCEqXS8PoH57XaizXKMTL_UQ/search?query=ict+elias
25. https://www.rijksictdashboard.nl/
26. https://www.scaledagileframework.com/

Reports on success/failure of IT-projects

Standish: Rapport 2015

Fujitsu: The Digital Transformation PACT (26 February 2018)

Professional Standards

1. Axelos, Management Successful Projects with PRINCE2, 2009 Edition Manual, The Stationary Office- Crown, UK, 2014
2. http://www.efqm.org/efqm-model/criteria/results
3. *"The BPM Profession - ABPMP International"*. *www.abpmp.org.* Retrieved 20 January 2018.
4. http://products.ipma.world/ipma-product/peb/read-peb/
5. https://www.pmi.org/pmbok-guide-standards/foundational
6. SDM System (1992(Cap Gemini Publishing)
7. http://iso25000.com/index.php/en/iso-25000-standards/iso-25010
8. IFPUG on function points and snap-points: http://www.ifpug.org/

Literature

1. Bayens, Tönissen, (2013), *Enterprise & Architecture*, Van Haaren Publishing
2. Bannerman, Paul, L, (2008) Defining Project Success: A Multilevel Framework, NICTA, Australian Technology Park, Sydney, Australia
3. Beck, Kent, James Grenning, Robert C. Martin, Mike Beedle, Jim Highsmith, Steve Mellor, Arie van Bennekum, Andrew Hunt, Ken Schwaber, Alistair Cockburn, Ron Jeffries, Jeff Sutherland, Ward Cunningham, Jon Kern, Dave Thomas, Martin Fowler, Brian Marick (2001). *"Manifesto for Agile Software Development".* Agile Alliance. Retrieved 14 June 2010.
4. Bouman, Egbert, *SmarTEST, slim testen van informatiesystemen*. SDU, 2008
5. Bradley, Gerald (2016)*, Benefits realisation management*. Routledge, Abingdon, USA
6. Bronsgeest, W.l. (2016), Meer vorm dan inhoud. Onderzoek naar evaluaties van ICT-projecten bij de overheid, Enschede NederlandDenning, P.J. (1997), *"A New Social Contract for Research",* Communications of the ACM (40:2) Febr 1997, pp. 132-134

7. Diest, B. van, (2009), *Can Anybody Help? Mitigating IS development project risk with user participation*. Master Thesis, University of Twente
8. Emery, F.E., Trist, E.L. (1960.), *Socio-Technical Systems. Management sciences, models and technique,* C.W and others Churchman. London: Pergamon.
9. M.G.J. van den Brand and J.F. Groote. *Software Engineering: Redundancy is Key.* Science of Computer programming 97:75-81, DOI: 10.1016/j.scico.2013.11.020, 2015.
10. L. Eveleens, J. (2011), *IT Forecast Quality in Measure and Number*, VU
11. J.F. Groote Modeling and Analysis of Communicating Systems, Groote/Mousavi, MIT Press 2014
12. Haverkate, J.M. (2011). *Opportunisme bij aanbestedingen inhuur ICT personeel.* Master's Thesis Report, Procurement MBA, Nijenrode Business University
13. Heemstra, F.J. (1994), Van kwaliteitsbehoeften naar kwaliteitseisen, Lansa Publishing, Leidschendam
14. Heemstra, F. J., & Heemstra, F. J. (1993). *Software cost estimation. Information and software technology, 34(10), 627-639*. DOI: 10.1016/0950-5849(92)90068-Z
15. Henderson, J.C,. Venkatraman, (1993), *Strategic Alignment: Leveraging Information Technology for Transforming Organizations,* IBM Systems Journal, volume 32, no 1
16. Hevner Alan R, et al, Design Science in Information Systems Research., MS Quarterly, Vol 28 No1, pp 75-105, March 2004
17. Computational Science and Its Applications - ICCSA 2006: International .
18. Kampman, J. (2015). *Van Stakeholdersanalyse naar Performancemangement,* Master's Thesis Report, Master of Science in Management, Open University
19. ISR (2002) *Editorial Statement and Policy*. Information Systems Research (13:3) December 2002
20. Kiel, W. (2010) *Een automatiseringsproject, een kwestie van alleen een systeem plaatsen?* Master's Thesis Report, Master of Crisis and Disaster Management (MCDM), Instituut voor Fysieke Veiligheid
21. Kools, T.M. (2013). *De precontractuele informatieplicht in Franchise overeenkomsten.* Master's Thesis Report, Master rechtsgeleerdheid, Privaatrecht, Tilburg University
22. Koorevaar, P and Noordam, (2010), *Business Process Management,* CAIP-reeks, Kluwer, Deventer
23. KPMG, (2017) *Agile project delivery How to increase project success in a hybrid world?* http://files.smart.pr/f7/455d40f28111e6a8a237660cbca18a/KPMG-2017-Survey---Agile-project-delivery.pdf
24. Kruchten, Philippe, The rational unified process: an introduction, - 3d ed.2004 Pearson Education Inc.
25. Kuper, R.G.J. (2009). *Business IT Alignment*, Master's Thesis Report, Postgraduate IT Audit, VU Amsterdam
26. Cristina Mele, Jacqueline Pels, Francesco Polese, (2010) A Brief Review of Systems Theories and Their Managerial Applications. Service Science 2(1-2):126-135. https://doi.org/10.1287/serv.2.1_2.126
27. Melville, N., Kraemer, K. and Gurbaxani, V. (2004). "Review: Information Technology and Organizational Performance: An Integrative Model of IT Business Value". MIS Quarterly, 28(2), pp. 283-322.
28. Van der Molen, "Waarom doen we dit eigenlijk? " van Duuren Management, 2013
29. Müller, (1969). *Woordenboek Grieks-Nederlands* Wolters-Noordhoff, Groningen, Amsterdam
30. Nesma, handboek telrichtlijnen V2.2 (2004)
31. NZA, (April 2015) Thematisch onderzoek aanpak PGB-fraude Onderzoek naar de effecten van de intensiveringsmaatregelen voor de aanpak van PGB-fraude door de zorgkantoren
32. Op 't Land, Martin (2008), *Applying Architecture and Ontology tot the Splitting and Allying of Enterprises*, TU Delft

33. A.A.H. Osaiweran. M. Schuts, J. Hooman, J.F. Groote and B. van Rijnsoever. Evaluating the effect a lightweight formal technique in industry. International Journal on Software Tools for Technology Transfer 18(1):93-108, 2016. DOI: 10.1007/s10009-015-0374-1.
34. Peters. Thomas J. and Waterman Jr, Robert H. (1982), *In Search of Excellence.* Harper & Row, Publishers, New York
35. Pressman, Roger S. (2005). *Software Engineering, A Practioner's Approach*, McGrawHill
36. Rengers, Merijn (1 maart 2016), *De tragische ondergang van voetbalsoftware*, NRC
37. Robins, Stephen P. and Timothy A. Judge (2011), *Organizational Behavior.* Pearson Education publishing as Prentice Hall
38. Rooks, G. (2002). *Contract en Conflict, Strategisch Management van Inkooptransacties*. Doctoral thesis at the faculty of Sociology, Utrecht University.
39. Senge, P.M., Sterman, J. 1990. *System Thinking and Organizational Learning: acting locally and thinking globally in the organization*. Working Paper n.03.009. Cambridge: MIT.
40. Slot, R.G., *A method for valuing Enterprise Architecture based Business Transformation and Measuring the value of Solutions Architecture*, University of Amsterdam, 2010
41. Soh, Christina and Markus, M. Lynne, 1995. *"How IT Creates Business Value: A Process Theory Synthesis"* . *ICIS 1995 Proceedings*. 4. https://aisel.aisnet.org/icis1995/4Sommerville, Ian (2001), Software Engineering, 6th edition Addison Wesley
42. Stamper, T. (2009). *Mislukte IT-projecten: een kwestie van plannen?* Bachelor's Thesis Report, Master of Information Science, Radboud University
43. Verhoef, C (2002) Quantitative IT portfolio management, Elsevier, Science of Computer Programming 45
44. Van Vossen, J.W. (2014), *Proactief juridisch handelen binnen de bouw als de nieuwe norm*, Master's Thesis Report, Master rechtsgeleerdheid, Ondernemingsrecht, Tilburg University
45. Vroonland, (2012), *Zwart-op-wit: contractgebruik in R&D-samenwerkrelaties*. Master's Thesis Report, Master Business Process, Management & IT,
46. Weinberg, Gerald. M. (1998*), The psychology of Computer Programming*, Dorset House Publishing, New York
47. Weinberg, Gerald M. (2011), *General Systems Thinking,* Smashwords (e-book)
48. de Wit, Bob and Ron Meyer (2010), *Strategy, Process, Content, Context, an international perspective,* South-Western Cengage Learning

APPENDIX 1 – Methods

1.1 Contextual

The DESTEP analysis

In this section the DESTEP analysis is described. DESTEP is an abbreviation for demographic, economic, social/cultural, technological, ecological and political/judicial factors.

This analysis enables organisations to get insight in their external (macro) environment. Insight into the external environment of an organisation is of importance in order to effectively make use of the opportunities and threats caused by the landscape wherein a business operates.

For IT projects it enables organisations to get a full view on their stakeholders.

Supporting methods as the SWOT analysis or the Porter-scheme are important for the business to decide in which area they want to improve themselves. However, they contribute less to the definition of stakeholder requirements than the DESTEP-analysis.

The macro-environment variables of DESTEP Analysis are as follows:

Demographic: Demographic factors are characteristics of the population.

Economic: Economic factors are characteristics that describe the economy.

Social-cultural: Social-cultural factors are characteristics of the culture and customers

Technological; Technological factors are characteristics of the development

Ecological: Ecological factors are characteristics of the physical environment

Political-judicial: Political-judicial factors are characteristics of government decisions

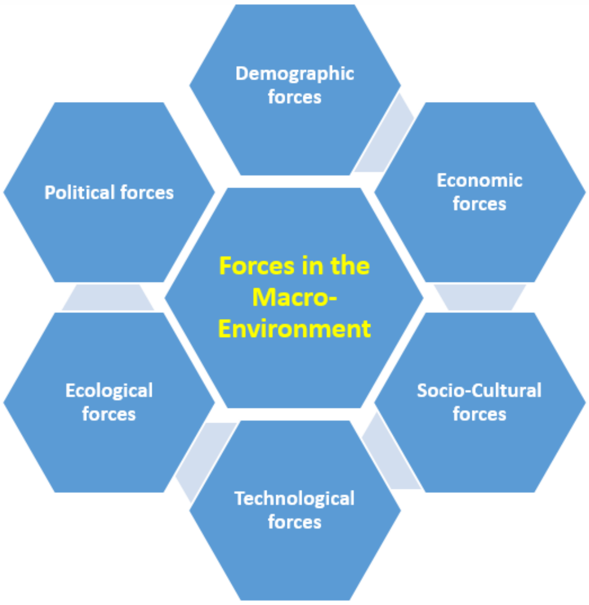


Figure 13

1.2 Enterprise

EFQM

(from the “EFQM Excellence Model, EFQM, 2013) “*The EFQM model (in the Netherlands known as INK model) The EFQM Excellence Model is based on a set of European values, first expressed in the European Convention on Human Rights (1953) and the European Social Charter (revised in 1996).”*
“The Fundamental Concepts of Excellence build on the foundation of these basic human rights, assuming they are universally applied. Recognising the role business can play in supporting the broader goals of the United Nations, the UN Global Compact (2000) was established. This initiative encourages organisations to actively apply these values, set out as 10 Principles for sustainable and socially responsible business, across their global operations.
Whilst a number of these principles are explicitly covered in the EFQM Excellence Model, a number are implicit, including those relating to human rights, corruption, bribery and forced labour, as these are already a legal requirement within Europe.
The EFQM Excellence Model assumes that an excellent organisation will respect and comply with the 10 principles of the UN Global Compact, regardless of whether legally obliged to do so or not.
The need for a Model Regardless of sector, size, structure or maturity, organisations need to establish an appropriate management framework to be successful. The EFQM
Excellence Model is a practical, non-prescriptive framework that enables organisations to:

- *Assess where they are on the path to excellence; helping them to understand their key strengths and potential gaps in relation to their stated Vision and Mission.*
- *Provide a common vocabulary and way of thinking about the organisation that facilitates the effective communication of ideas, both within and outside the organisation.*
- *Integrate existing and planned initiatives, removing duplication and identifying gaps.*
- *Provide a basic structure for the organisation’s management system.*

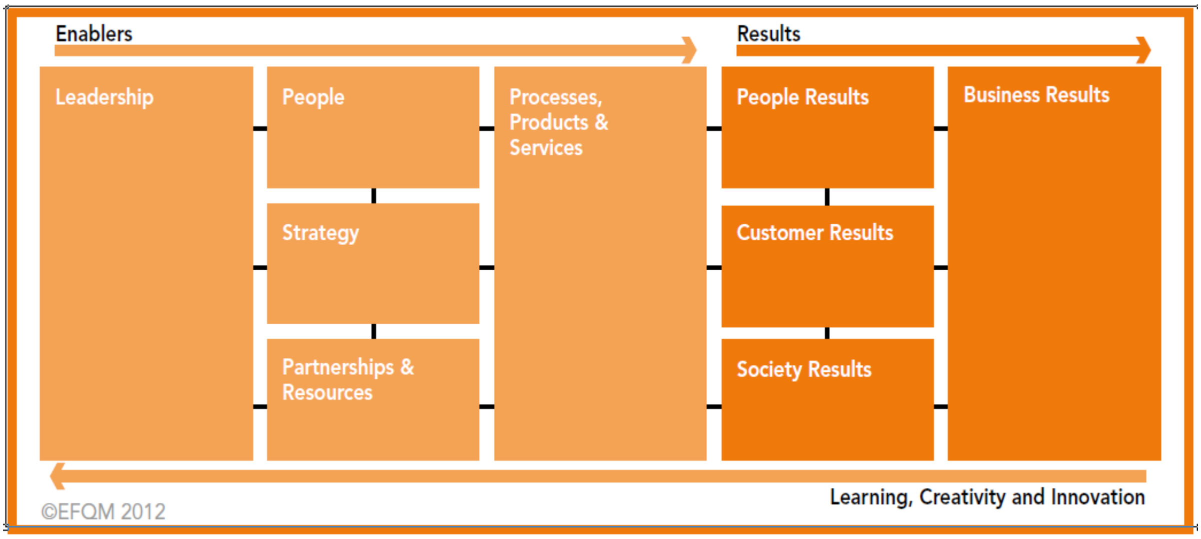


Figure 14

This is realised through a set of three integrated components which comprise the EFQM Excellence Model:

1. *The Fundamental Concepts of Excellence: The underlying principles which are the essential foundation of achieving sustainable excellence for any organisation.*

2. *The EFQM Excellence Model: A framework to help organisations convert the Fundamental Concepts and RADAR logic into practice.*
3. *RADAR logic: A dynamic assessment framework and powerful management tool that provides the backbone to support an organisation as it addresses the challenges it must overcome if it is to realise its aspiration to achieve sustainable excellence*."

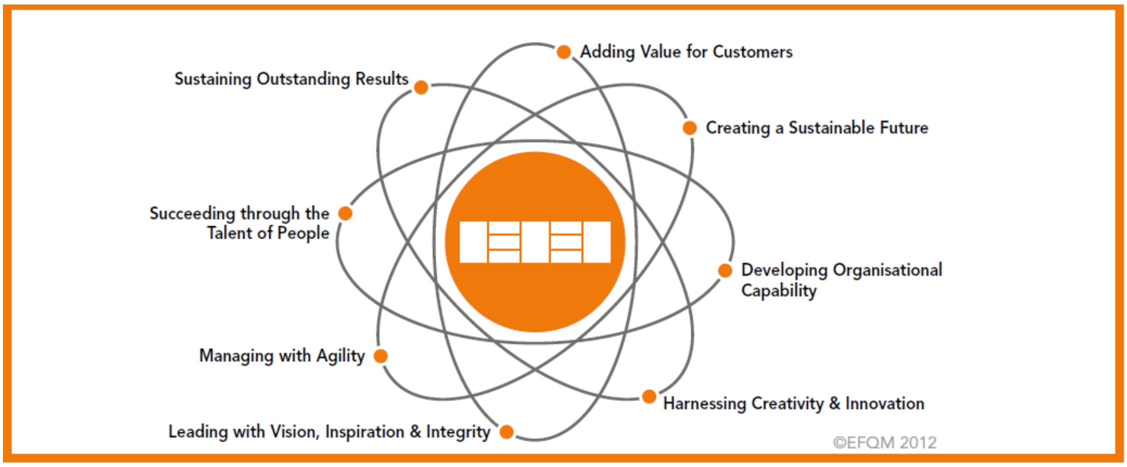

Adding Value for Customers
Sustaining Outstanding Results
Creating a Sustainable Future
Succeeding through the Talent of People
Developing Organisational Capability
Managing with Agility
Harnessing Creativity & Innovation
Leading with Vision, Inspiration & Integrity
©EFQM 2012

Figure 15

This model not only supports the mission and vison analysis, the analysis of the stakeholders, it also supports the next steps: it explicitly puts people and processes in the model.

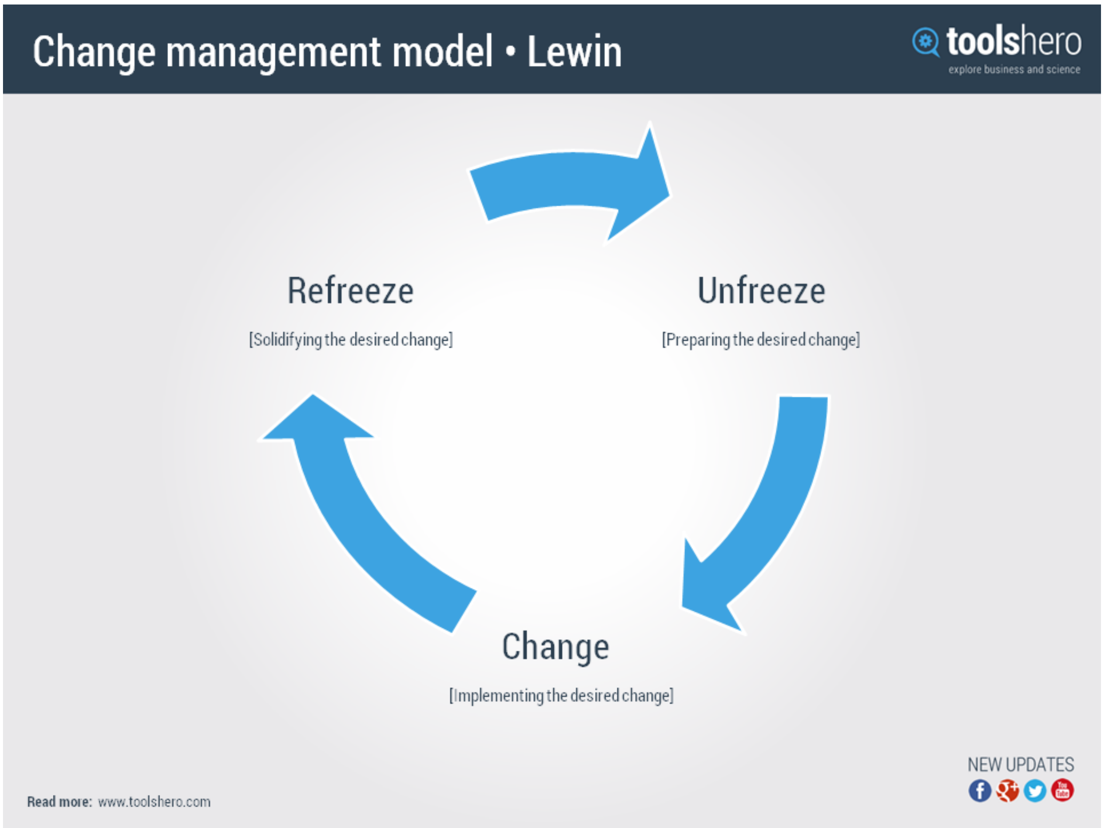
1.3 People

To manage the change two prevailing frameworks are used. Of course a lot more knowledge is necessary and available in the realm of change management. In this research only those are mentioned that support project management and the requirements gathering.

It is important to realise that stakeholders, managers at all levels are undergoing the same processes, since they are human too.

Lewin's change model versus behaviour modification

(With regards to Toolshero.com) Lewin's change model focuses on behaviour modification of people/ employees. The model distinguishes three stages.



Change management model • Lewin

toolshero
explore business and science
Refreeze
[Solidifying the desired change]
Unfreeze
[Preparing the desired change]
Change
[Implementing the desired change]
Read more: www.toolshero.com
NEW UPDATES

Unfreeze stage

Figure 16

At this stage people realize that something is going to change and they are dealing with strong emotions such as denial, impatience, uncertainty and doubt. It is essential for a business to fully disclose the state of affairs and to explain why a change process is put into force. As a result of clear communication employees are more willing to accept to the new change of direction and they can let go of old customs. During this stage, it is advisable to involve employees so that they can take a constructive approach to the change process.

Change stage

It is of crucial importance that the change is implemented within a short time. The longer the change process takes, the more employees are inclined to relapse into old habits and rituals. This stage is sometimes referred to as the “move stage” because it causes a ripple effect within an organization. By acting vigorously and implementing the change within a short time, the employees will become aware of the importance of this change more quickly.

Refreeze stage

This stage, refreeze the change, is about solidifying the change. After the change has been implemented in the change stage, employees are inclined to revert to their old habits. Therefore, it is advisable to make proper arrangements and carry out interim evaluations, monitor and make adjustments (when necessary). Only then the new situation can be stabilized and the employees will understand that there is no turning back. Eventually, they will act in accordance with the new situation and realize that there are advantages to it.

Kotter's Eight Step Plan

figure 17

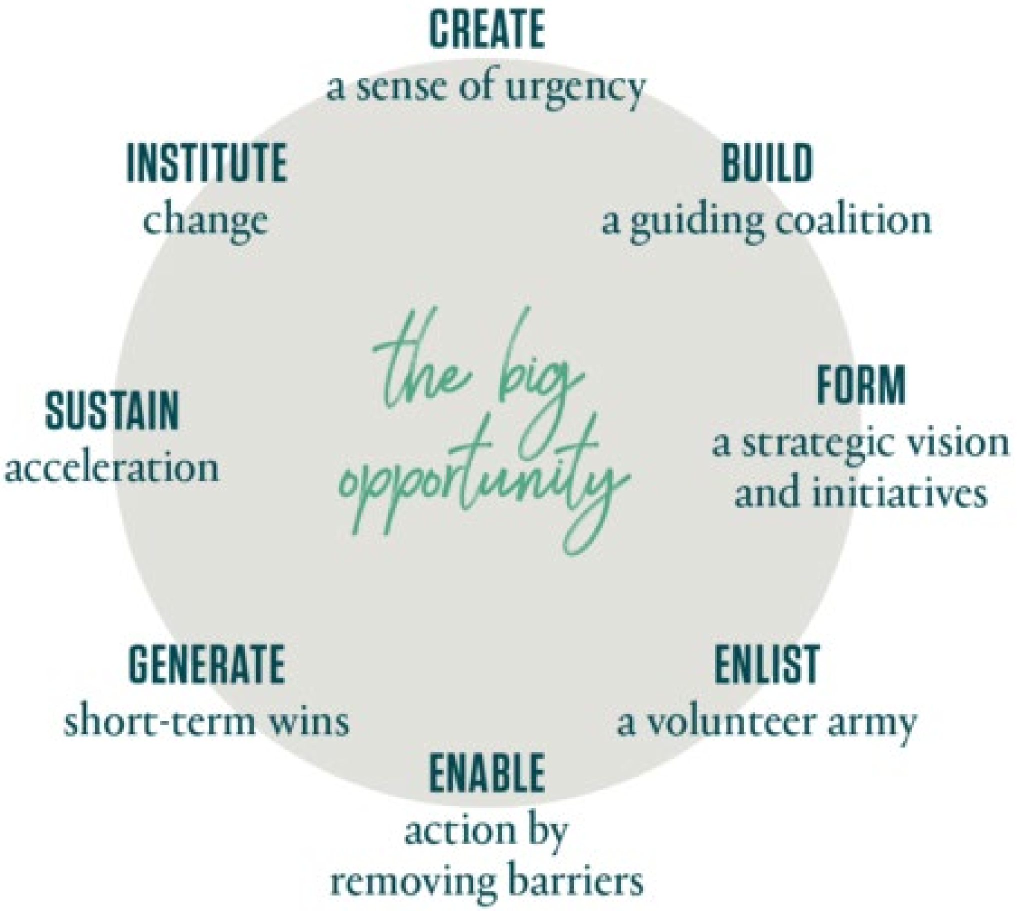


figure 17

(with regards to https://www.kotterinc.com); In this appendix we describe shortly the eight steps that support sustainable change:

Create a sense of urgency

Help others see the need for change through a bold, aspirational opportunity statement that communicates the importance of acting immediately.

Build a guiding coalition

A volunteer army needs a coalition of effective people – born of its own ranks – to guide it, coordinate it, and communicate its activities.

Form a strategic vision and initiatives

Clarify how the future will be different from the past and how you can make that future a reality through initiatives linked directly to the vision.

Enlist a volunteer army

Large-scale change can only occur when massive numbers of people rally around a common opportunity.
They must be bought-in and urgent to drive change – moving in the same direction.

Enable action by removing barriers

Removing barriers such as inefficient processes and hierarchies provides the freedom necessary to work across silos and generate real impact.

Generate short-term wins

Wins are the molecules of results. They must be recognized, collected and communicated – early and often – to track progress and energize volunteers to persist.

Sustain acceleration

Press harder after the first successes. Your increasing credibility can improve systems, structures and policies. Be relentless with initiating change after change until the vision is a reality.

Institute change

Articulate the connections between the new behaviors and organizational success, making sure they continue until they become strong enough to replace old habits.

Requirements from HR

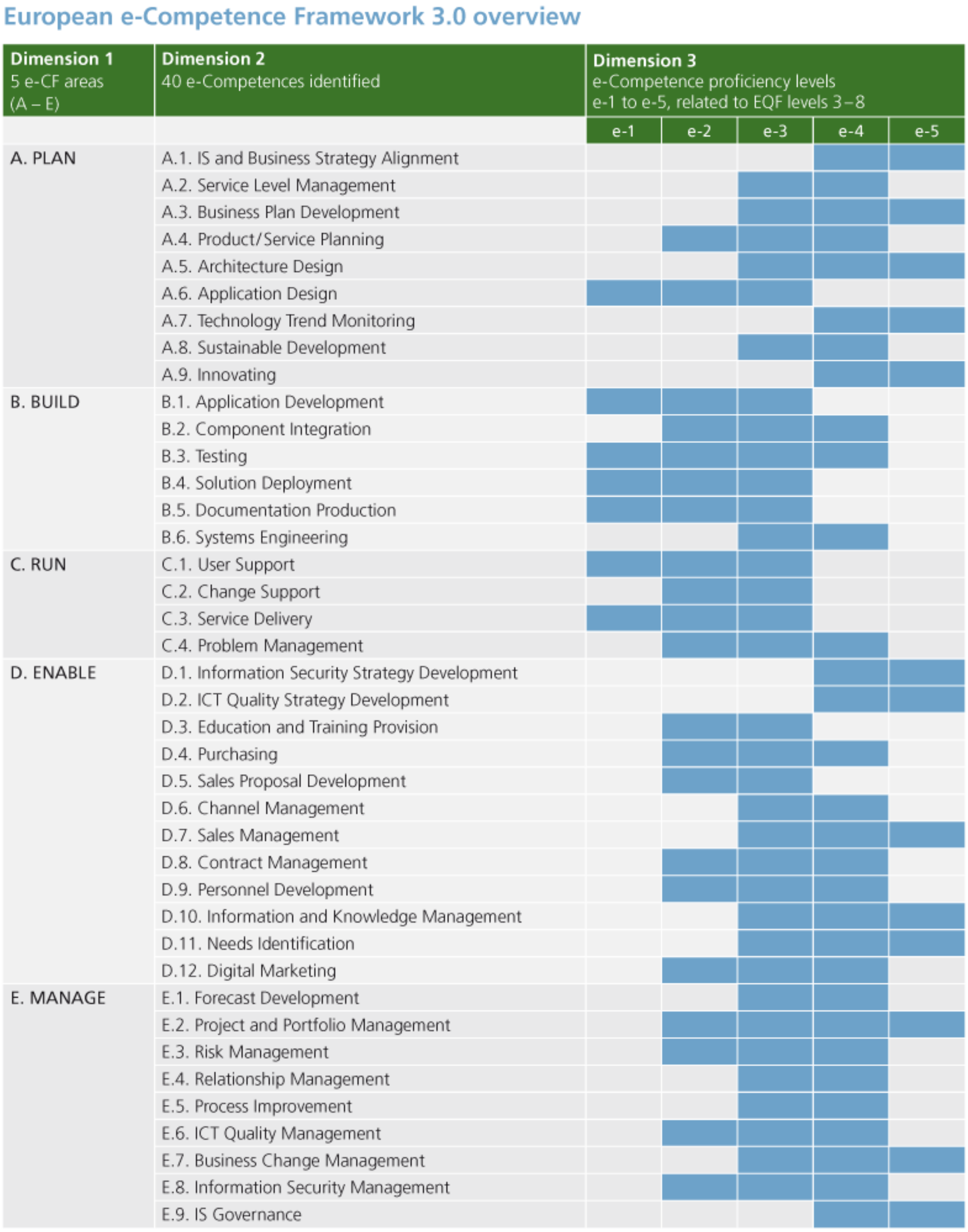
In this appendix the main categories are mentioned. In each enterprise, depending on the appointments with the workers' council and collective agreements specific requirements are to be considered.

ISO25010

In ISO25010 especially the requirements on usability and quality in use (satisfaction) are important, besides of course the functional requirements, see APPENIDX 3

Competence management

To decide which digital competences are necessary after the change , the e-CF model (EU) is a good guide. See http://www.ecompetences.eu/get-the-e-cf/ for the full version:



European e-Competence Framework 3.0 overview

Dimension 1 5 e-CF areas (A – E)	**Dimension 2** 40 e-Competences identified	**Dimension 3** e-Competence proficiency levels e-1 to e-5, related to EQF levels 3–8				
		e-1	e-2	e-3	e-4	e-5
A. PLAN	A.1. IS and Business Strategy Alignment				■	■
	A.2. Service Level Management			■	■	
	A.3. Business Plan Development			■	■	■
	A.4. Product/Service Planning		■	■	■	
	A.5. Architecture Design			■	■	■
	A.6. Application Design	■	■	■		
	A.7. Technology Trend Monitoring				■	■
	A.8. Sustainable Development			■	■	
	A.9. Innovating				■	■
B. BUILD	B.1. Application Development	■	■	■		
	B.2. Component Integration		■	■	■	
	B.3. Testing	■	■	■	■	
	B.4. Solution Deployment	■	■	■		
	B.5. Documentation Production	■	■	■		
	B.6. Systems Engineering			■	■	
C. RUN	C.1. User Support	■	■	■		
	C.2. Change Support		■	■		
	C.3. Service Delivery	■	■	■		
	C.4. Problem Management		■	■	■	
D. ENABLE	D.1. Information Security Strategy Development				■	■
	D.2. ICT Quality Strategy Development				■	■
	D.3. Education and Training Provision		■	■		
	D.4. Purchasing		■	■	■	
	D.5. Sales Proposal Development		■	■		
	D.6. Channel Management			■	■	
	D.7. Sales Management			■	■	■
	D.8. Contract Management		■	■	■	
	D.9. Personnel Development		■	■	■	
	D.10. Information and Knowledge Management			■	■	■
	D.11. Needs Identification			■	■	■
	D.12. Digital Marketing		■	■	■	
E. MANAGE	E.1. Forecast Development			■	■	
	E.2. Project and Portfolio Management		■	■	■	■
	E.3. Risk Management		■	■	■	
	E.4. Relationship Management			■	■	
	E.5. Process Improvement			■	■	
	E.6. ICT Quality Management		■	■	■	
	E.7. Business Change Management			■	■	■
	E.8. Information Security Management		■	■	■	
	E.9. IS Governance				■	■

flgure 18

1.4 Usage

Usage consist of processes and usability.

Processes

As the quote on the website of the ABPMP (https://www.abpmp.org/page/BPM_Profession) *"ABPMP International considers Business Process Management (BPM) to be both a management discipline and a set of technologies that support managing by process. Unfortunately, that seems to be the only industry consensus on the definition of BPM, as there is certainly no shortage of opinion about "WHAT" BPM is and "HOW" to do BPM."*

The "official" statement here is that no uniformity exists in process management. Most methods are depending on ho specific consulting firms are implementing this.

Tooling (of which a wide variety also exists) somewhat unifies the methods by improving their exchangeability. Process redefinitions provides requirements for the use cases or user stories and their impact can provide input for the project plan.

The main "agreed" steps are:

By Aleksander Blomskøld - Own work, CC BY-SA 3.0,

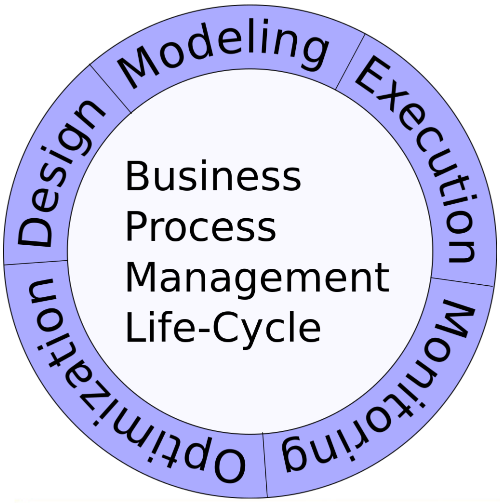


Figure 19

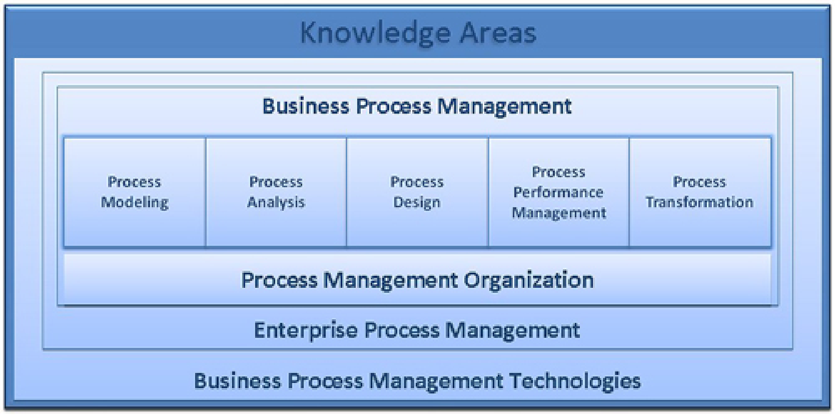

Knowledge Areas
Business Process Management
Process Modeling
Process Analysis
Process Design
Process Performance Management
Process Transformation
Process Management Organization
Enterprise Process Management
Business Process Management Technologies

Figure 20

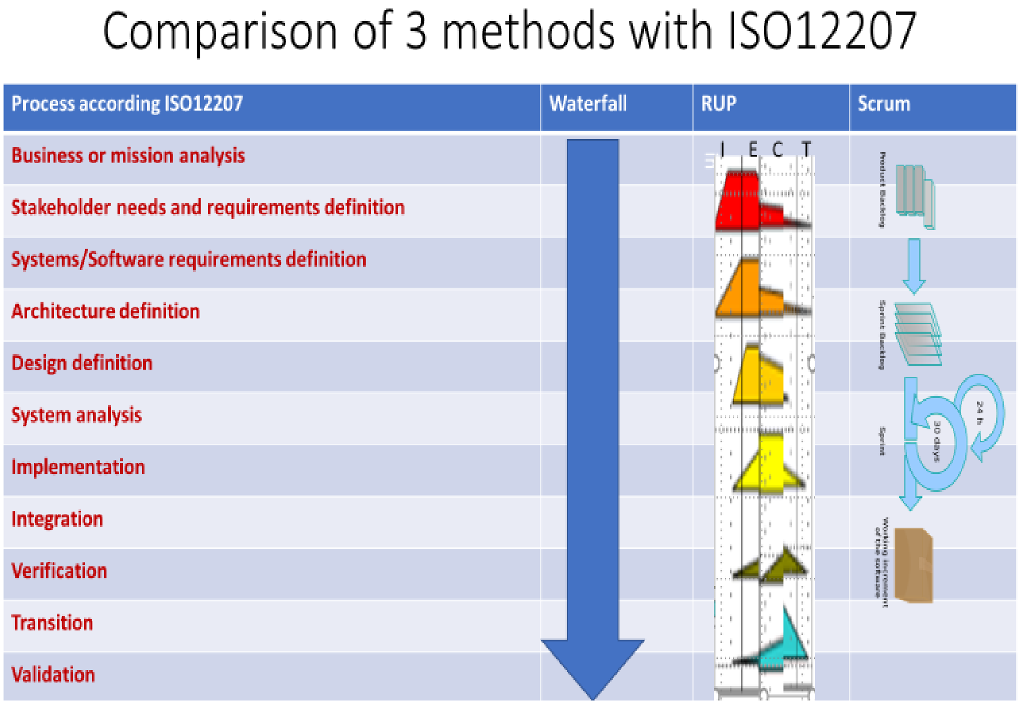
Usability

Usability requirements are provided by https://www.w3.org/standards/

And in the Netherlands by https://www.digitoegankelijk.nl/beleid/wat-is-verplicht

1.5 IT Methods

All methods are compared to ISO 12207. The table below shows the comparison t0 12207 in this appendix we elaborate the methods and show their analogies with ISO12207



Comparison of 3 methods with ISO12207

Process according ISO12207	Waterfall	RUP	Scrum
Business or mission analysis			
Stakeholder needs and requirements definition			
Systems/Software requirements definition			
Architecture definition			
Design definition			
System analysis			
Implementation			
Integration			
Verification			
Transition			
Validation			

I E C T
Product Backlog
Sprint Backlog
Sprint
30 days
24 h
Working increment of the software

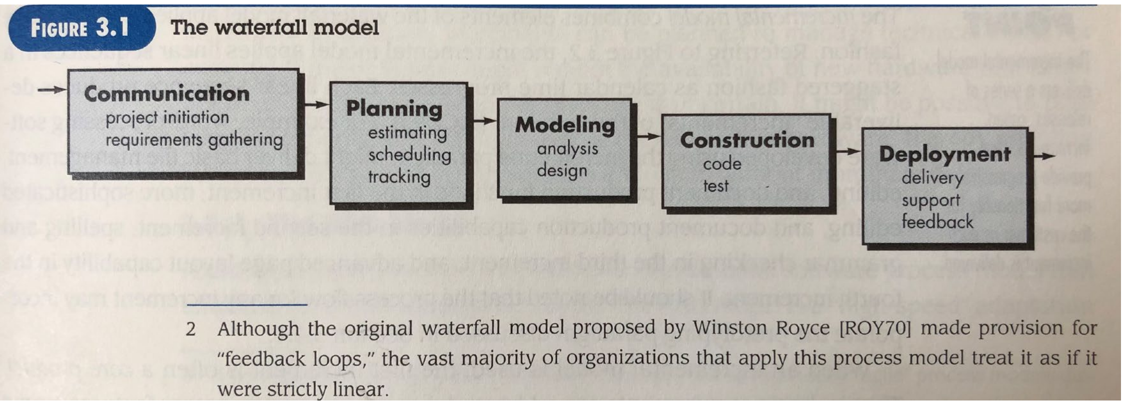
Table 2

Waterfall

Copied from Pressman (2005)

Waterfall (or the Classic Life Cycle) is developed in the time mainframes existed and programming took time and was not well supported by tooling. The original model was proposed by Winston Royce in 1970). In the Netherlands SDM was seen as a waterfall method. Actually that is misunderstood: it was a set of best practices in processes and activitities: in the last chapter various ways of working were proposed, depending on the situation.

FIGURE 3.1 The waterfall model



2 Although the original waterfall model proposed by Winston Royce [ROY70] made provision for "feedback loops," the vast majority of organizations that apply this process model treat it as if it were strictly linear.

Figure 21 from Pressman

Process		
	Activity	ISO12207
Communication		
	Project initiation	Technical management
	Requirements gathering	Business or mission analysis Stakeholder needs and requirements definition Systems/Software requirements definition
Planning		
	Estimating	Technical management
	Scheduling	Technical management
	Tracking	Technical management
Modeling		
	Analysis	System analysis
	Design	Design definition
Construction		
	Code	Implementation
	Test	Verification
Deployment		
	Delivery	Transition Maintenance
	Support	Operation
	Feedback	Validation

Table 12

That waterfall is developed in the tome of mainframe systems appears clearly from the fact that architecture and integration (in ISO12207) is not present.

RUP

RUP or UP is developed by Kruchten. Booch, Rumbaugh, and Jacobson developed at the same time the “Unified Modelling Language”. The method is related to the object oriented way of programming that originated form 1990.

RUP was owned by Rational, that developed tools for UML. Rational has been bought by IBM. After a debate the method is handed over to the Open Group and now (and then) is called UP again.

The method is characterized by three pillars: Phases, Disciplines and Iterations.

Phases are supporting decision making by project management and project boards and resembles the technical management processes of ISO12207. When a phase is passed it is “forbidden” to go back. This means that thorough inspection per phase should be executed.

- Inception : after this phase the Why of the Project is clear, and the technical solution is proved to be feasible.
- Elaboration: after this phase the software architecture proves to be right and the most difficult parts function.
- Construction: building, integrating and testing the iterations
- Transition: deployment of the total solution, after acceptance tests.

Iterations are delivering working entities, using disciplines a seen in the picture.

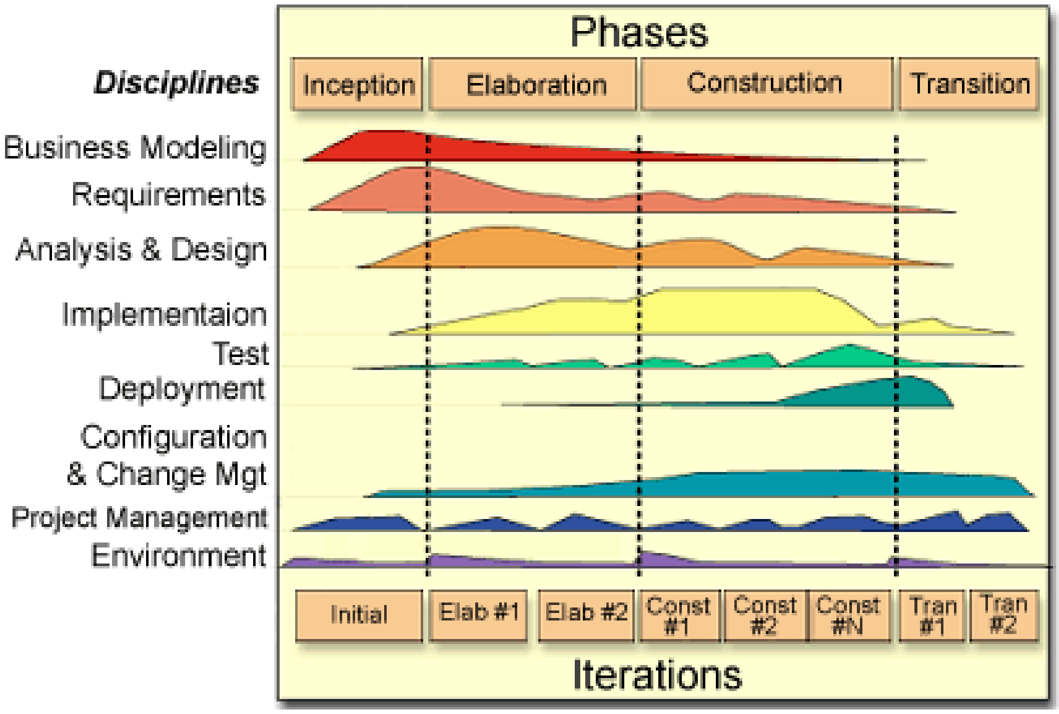


Figure 22

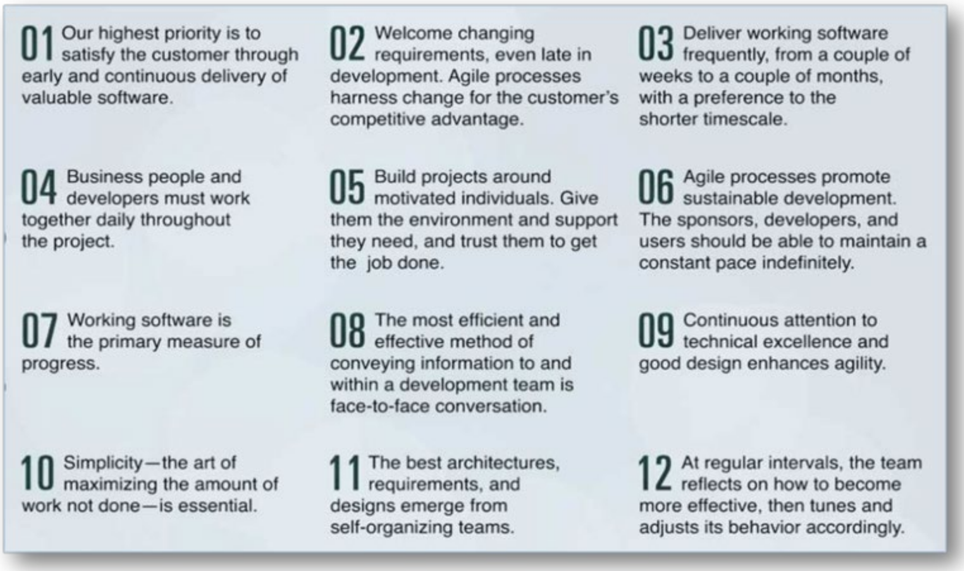
For this purpose a comparison of the disciplines with ISO12207 is made.

Discipline	ISO12207
Business modelling	Business or mission analysis
Requirements	Stakeholder needs and requirements definition Systems/Software requirements definition
Analysis and Design	Architecture definition
	System analysis
	Design definition
Implementation	Implementation
	Verification
Deployment	Transition
	Validation

Table 13

Agile/Scrum

The agile alliance defined 12 principles for Agile software development in the Agile Manifesto



01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

04 Business people and developers must work together daily throughout the project.

05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

07 Working software is the primary measure of progress.

08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

09 Continuous attention to technical excellence and good design enhances agility.

10 Simplicity—the art of maximizing the amount of work not done—is essential.

11 The best architectures, requirements, and designs emerge from self-organizing teams.

12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 23

Scrum is one of the Agile methods used regularly. Scrum is a process model, not a development model that creates an answer to the unpredictability of requirements. The flow as found at scrum.org

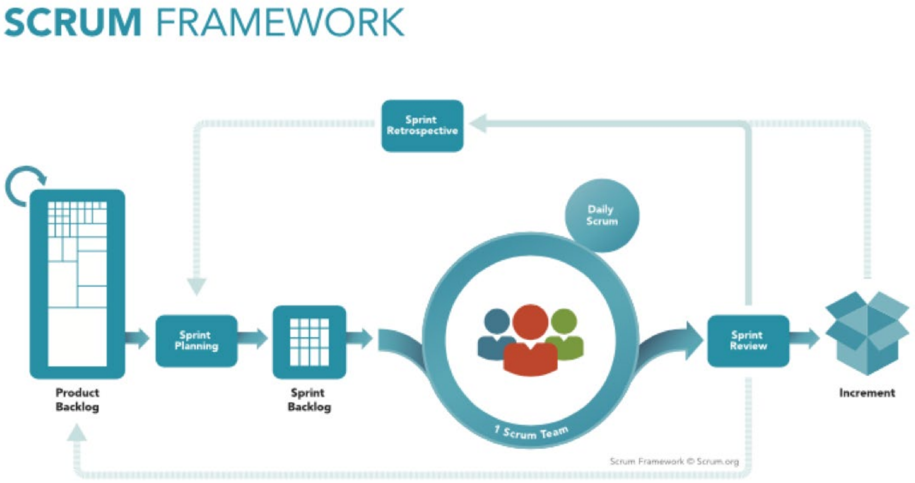


Figure 24

Process	ISO12207
Product Backlog	Technical Management (+ requirements?)
Scrum Planning	Technical Management
Sprint Backlog	Technical Management
Scrum team	All software / technical processes
Daily Scrum	Implementation
Sprint Review	Verification
Increment	Transition
Sprint Retrospective	Technical Management

Table 14

APPENDIX 2 - ISO25010

Typology	*attribute*	description
Product Quality Extent to which software product or software product meets purpose.		
Functional suitability		
delivers functions that corresponds to the specified and assumed needs, when used under specified conditions	*Functional completeness*	supports all tasks and goals
	Functional correctness	provides accurate results
	Functional appropriateness	supports specific tasks and goals
Performance efficiency		
output in relation to used resources when used under specified conditions	*Time-behaviour*	response, processing and throughput meets requirements
	Resource utilization	use of quantity and type of resources meets requirements
	Capacity	the maximum limits of the parameters of product/system meets/trespasses wishes
Compatibility		
The extent to which an application or component is able to exchange information with other products/systems/components and or is able to perform when sharing the same infrastructure environment	*Co-existence*	efficiently performs its functions while sharing infrastructures and resources, without detrimental effects on other products
	Interoperability	can exchange usable information with other systems, products or companies
	attribute	description
Usability		
can be used satisfactorily by specific users to reach their goals	*Appropriateness recognisability*	can be valued by the users to ensure that it meets their needs

efficiently and effectively in a specified user context		
	Learnability	can be taught with specific learning goals for the specified users to learn to use it satisfactorily, effective, efficient with freedom risk in the specified context.
	Operability	contains attributes to operate and control it
	User error protection	protects users against errors /prevents errors
	User interface aesthetics	has a user interface that provides a smooth and satisfying interaction
	Accessibility	includes every user, regardless of a specific disability to reach a specified goal in a specified usage context
	Reliability	performs under specified conditions within a given amount of time
	Maturity	or component is reliable under normal conditions
	Availability	is ready to use when needed
	Fault tolerance	can function even when hardware- or software errors occur.
	Recoverability	when interrupted or stopped by errors, can recover the data to return to the prior functioning state.

	attribute	description
Security		
protects information and data in order to provide access to persons, products or systems at their specific level of authorisation	*Confidentiality*	secures that data is only accessible (CRUD) for those who are authorised
	Integrity	prevents unauthorised access or adjustment of applications and data
	Non-repudiation	proves that actions or events have taken place, to prevent denial
	Accountability	Actions of an entity can be traced to that entity
	Authenticity	can prove the identity of a source or subject
Maintainability		
can be adapted effective and efficient by the appointed functional and technical system managers	*Modularity*	is composed of loosely coupled modules to ensure that changes in one module has the least possible impact on others.
	Reusability	produces components that can be used in other systems or as component of a new one
	Analysability	provides insight in how to change it effective and efficient ad the ease to detect errors/malfunctioning
	Modifiability	can be modified effective and efficient without noticeable impact (no errors, same or higher quality)
	Testability	Acceptance criteria can be declared and executed to prove that the criteria are met
Portability		
can be ported to another infrastructure environment	*Adaptability*	can be adapted when the infrastructure is changing
	Installability	can be easily installed or removed from a specified environment
	Replaceability	can be replaced by another system with the same purpose.
Typology	*attribute*	description
Quality in Use		

Effectiveness ***Accuracy and completeness to reach specified targets by users***		
Efficiency ***Needed resources are in proportion to the required effectiveness***		
Satisfaction		
The extent to which user needs are fulfilled when the product or system is used in a specified context	*Usefulness*	The extent to which a user is satisfied with reaching the by user observed targets and implications of using the system
	Trust	The extent to which an user or other stakeholders trust the behaviour of the system
	Pleasure	The extent to which an user is pleased to fulfil his personal needs
	Comfort	The extent to which an user is satisfied with his physical well-being
Freedom from risk		
The extent to which the product or system mitigates the risks concerning finance, efficiency, properties, reputation or other resources in the context of the use of this product or system	*Health and safety risk mitigation*	The extent to which a product or system mitigates the risk for humans in the context of use
	Environmental risk mitigation	The extent to which a product or system mitigates potential risks concerning properties or the environment in the context of use
Context coverage		
The extent to which a product or system can be used effectively, efficiently, risk free and satisfactory as well in the specified context of use as in initially not defined contexts of use	*Context completeness*	The extent to which a product or system can be used effectively, efficiently, risk free and satisfactory in all defined contexts of use
	Flexibility	the extent to which a product or system can be used effective, efficient, risk free and satisfactory in contexts that are not initially specified in the requirements

Table 15

APPENDIX 3 - Survey

Questionnaire (check https://nl.surveymonkey.com/home/?ut_source=header)

Nr	Question	Multiple Choice	#
1	Mijn organisatie	heeft (overwegend) een publieke taak	14
		heeft (overwegend) een maatschappelijke taak	10
		heeft (overwegend) een winstoogmerk	14
		Overige (geef nadere toelichting)	-
2	Uw rol in het programma/project wordt het best omschreven als	Senior Responsible Owner	17
		Voorzitter Stuurgroep/Program Board	1
		Senior Responsible User	12
		Lid van de stuurgroep als vertegenwoordiger van een groep stakeholders	1
		Anders, namelijk project/programmamgr	7
3	Uw programma/project:	Is al geïmplementeerd	7
		Is gedeeltelijk geïmplementeerd	19
		Is (nog) niet geïmplementeerd	12
		zal het ook niet meer worden	-
4	De begrote kosten voor het ICT gedeelte bedragen	Minder dan 1 miljoen	17
		Tussen 1 en 5 miljoen	10
		5 miljoen of meer	9
		onbekend	2
5	Voor de bestedingen ten opzichte van de begroting geldt:	bestedingen zijn significant lager dan begroting (binnen de marge van 20% van het begrote bedrag	1
		bestedingen zijn binnen de marge van 20% van het begrote bedrag	25
		de bestedingen zijn 20-50% hoger	4
		de bestedingen zijn meer dan 50% hoger dan de prognose	3
		Zou ik moeten navragen	5
6	De belangrijkste reden om dit programma/project uit te voeren is	Verdere digitalisering producten/diensten tbv burger, klanten, leveranciers	15
		Nieuwe bedrijfsfunctie (nieuwe wet/wetswijziging, nieuwe dienstverlening)	5
		Fusie of reorganisatie	2
		Verbeteren bedrijfsvoering	11
		Vernieuwen applicatielandschap/infrastructuur met het oog op de toekomst	5
		Verlagen kosten	-
		Overige (geef nadere toelichting)	-
7	De doelen die in de business case gesteld zijn	Worden geheel behaald	9
		Worden gedeeltelijk behaald, maar blijft relevant positief	26

		Worden niet (meer) behaald	2
		Zou ik moeten nakijken	1
8	De looptijd van het programma/project is	minder dan 1 jaar	11
		1 tot 3 jaar	19
		3 tot 5 jaar	5
		langer	3
9	De technologische component van dit programma/project bestaat voor 80% uit	Maatwerk programmatuur	14
		Gebruik en integratie van standaard open componenten	8
		Een standaard oplossing (pakket)	9
		Kantoortoepassingen	1
		Hardware en netwerk	3
		Overige (geef nadere toelichting)	3
10	Programma/projectmanager	Is een interne medewerker	16
		Is een ervaren ZZP'er	14
		is in dienst van een IT/consultancy bedrijf van minder dan 100 werknemers	4
		Is in dienst van een tt de top 5 behorend IT/consultancy bedrijf	2
		Overige (geef nadere toelichting)	2
11	De opdracht relatie met de uitvoerders (ontwerp/bouw) is al volgt	We werken voornamelijk met interne medewerkers (10-20% inhuur)	14
		We huren mensen in, maar sturen zelf aan	12
		We huren expertise in en verwachten dan voorstellen en feedback op onze plannen	1
		We laten de opdrachtnemer uitwerken volgens onze specificaties en controleren op output	3
		We leggen de verantwoordelijkheid bij de opdrachtnemer voor een geslaagde implementatie in de organisatie	4
		We verwachten van de opdrachtnemer dat de business case gerealiseerd worden en sturen daar dus nauwelijks in mee	2
		Zou ik moeten nakijken	1
		Overige (geef nadere toelichting)	1
12	Kwaliteitsmanagement	Onze organisatie verzorgt zelf het kwalteitsmanagement over en dus buiten het project/programma	14
		Een specialist (ZZP/gespecialiseerd bedrijf) verzorgt het kwaliteitsmanagement en rapporteert aan programma/project en MT	1
		Kwaliteitsmanagement wordt uitgevoerd wanneer er problemen dreigen/zijn	1
		Kwaliteitsmanagement vindt plaats binnen het programma	13
		Kwaliteitsmanagement is niet expliciet belegd	9

13	Doelstellingen van het programma/project	De doelstellingen zijn vanaf het begin SMART verwoord	9
		We hebben doelstellingen, die zijn in de voorbereidingsfase concreet gemaakt	24
		We hebben bewust geen doelstellingen geformuleerd: we ontwikkelen aan de hand van de hoogste business value	4
		Niet nodig, de wet schrijft voor	-
		Hebben we niet samen uitgesproken maar zijn er uiteraard wel	1
14	De gekozen oplossing	Vooraf zijn diverse scenario's geschetst, op hoofdlijnen, waar we uit gekozen hebben	19
		Er zijn vooraf gedetailleerde scenario's uitgewerkt zodat de mate waarin voldaan werd aan de business case helder was	9
		De gedetailleerde scenario's zijn aangevuld met migratie-scenario's	1
		Er zijn vooraf geen alternatieve scenario's aangegeven, die zijn juist tijdens de ontwerpfase aangegeven	5
		Er is één oplossing voorgesteld als de enige mogelijkheid, andere scenario's zijn niet getoond	2
		Overige (geef nadere toelichting)	2
15	Ontwerp van de processen	De processen zijn vooraf nieuw ontworpen omdat we vernieuwende technologie gebruiken	6
		Bestaande processen waren leidend en daarbij is de passende technologie gekozen	7
		Er zijn geen expliciete overwegingen gemaakt over al dan niet technologie maar de processen zijn wel vooraf in kaart gebracht	8
		De procesontwerpen worden tijdens de realisatie ontworpen/aangevuld	11
		De processen worden tijdens de implementatie met de gebruikers opgesteld	3
		Is mij niet bekend	1
		Overige (geef nadere toelichting)	2
16	De architectuur/het ontwerp	De oplossingsalternatieven zijn onderbouwd door en via de architectuur	15
		de architectuur is vervaardigd tijdens het plan/de roadmap	11
		de architectuur is vervaardigd toen de realisatie startte	2
		de architectuur is tijdens de uitvoering vervaardigd	3
		er is geen overall doel architectuur opgesteld	6
		Overige (geef nadere toelichting)	1

17	Andere eisen waar ik vooraf expliciet rekening mee heb laten houden (meerdere antwoorden mogelijk) Aanalhits van de 38 respondenten	Business continuïteit (welke onderbreking is maximaal toegestaan welke maatregelen zijn voorzien (via BIA)	26
		Toegankelijkheid (leesbaar voor mensen met welke beperking dan ook)	8
		Beveiliging	23
		Privacy (PIA)	21
		Performance (de benodigde snelheid/minimale wachttijd voor de eindgebruiker)	19
		Gebruiksgemak voor de interne medewerkers	23
		Beheerkosten	14
		Bij oplossingen die door derden worden gemaakt: overdraagbaarheid naar andere partij(en)	9
		Gebruik van open source oplossingen	5
		Overige (geef nadere toelichting)	6
18	Methode van werken	Ik heb/ de stuurgroep/programboard heeft bewust gekozen voor een bepaalde aanpak	28
		de keuze is overgelaten aan de professionals	8
		de leverancier bepaalde de methode	2
		ik weet het niet	-
19	Voor het project/programma management hanteren we	Prince2	10
		Ipma	-
		PMi	1
		De afspraken van de organisatie/het eigen handboek	17
		De werkwijze van de leverancier	5
		Onbekend	5
20	Als systeemontwikkelings-methode hanteren we	De afspraken van de organisatie/het eigen handboek	6
		Agile/Scrum	17
		Agile/Scrum in SAFe	1
		DSDM	-
		Andere Agile/Lean aapak	2
		RUP	-
		Waterval lineair	2
		een bewuste mix	3
		Niet bekend	7
21	Wat vond u van deze vragenlijst?	helder en begrijpelijk	25
		uitdagend	10
		te ingewikkeld	3
		Niet ingevuld	1

APPENDIX 4 - Justification of the modelling

Level	IS012207	Questions to ask						
		PLAN	DO	CHECK	Supplier or	Competences	Reasoning	Professional Deliverables
		the business proposition	creating the effect	validation of the solution	P U S E C			
	project management	Time constraint or budget constraint?	are milestones met according to budget?	Delivered on time and expectations met?				
Contextual	Business and Mission Analysis	How does this project contribute to the Mission and Vision of the organisation? Can you describe the goal of this project in one sentence? Is personal data involved? DPIA	Do all project activitities contribute to this same goal?	Are KPI's in place to measure the value and is the system providing the information for it?		Directing	The busines requirements are derived from the (new) company's mission and vision. The project has to be aligned and if not this must be resolved first.	Enterprise architecture Business architecture Project Plan Busines Validation
	Stakeholders needs Requirements definition	Which external stakeholders are involved in this change and in which role ? How are they committed to the outcome?	Are all stakeholders contributing according to their role? If not, does it need to be solved?	Do all stakeholders agree with the outcome, according to their role?		Strategic Management, Marketing, Public Affairs, Legal	For each stakeholder has to be defined what his role is, which information they need and which information or validation the projet needs	Business Architecture Information Architecture Business Validation Project Plan
	Stakeholders needs Requirements definition	Which new or changed products and services will be delivered to each stakeholder? Do these services and products fit in the mission and vision? Which value do they deliver to each of the stakeholders?	When other products and services are added, do you agree? Do they deliver extra value? Whta is the effect on the due date of the project?	Do the products and services deliver the intended value?		Marketing, Public Affairs, Legal, Finance and Control Security and Privacy Authority	Products and services contain information (data): producing them or receiving them needs business continuity measures and compliance to legislation. Adding new products or services during the project should be approved, to ensure legislative and busines continuity actions are taken.	Business Architecture Information Architecture DPIA's and BIA's first level Business Process (Re)design Business Use Cases
	Systems/Software requirements	Is digital exchange of data needed? How dependent are others or this organisation of this exchange? Are specific IT resources needed outside your organisation (hardware, software, people)?	Are the necessary contracts ready to use (and is risk mitigation in these contracts adressed)?	Can the project prove that these conditions are met at the given conditions?		Marketing Business Process Management IT department Risk & Control Purchasing Security and Privacy Authority	When parties depend on each other through their data exchange, business continuity of the exchange partners could be influenced. The cost of minimising the risk should be handled in the contracting. For IT contracts legal tender procedures could be needed which costs time. Sometimes money is only provided when targets at stages are met during the project.	Business Architecture Information Architecture ISO25010 Project Plan / Risk List Project Plan / Planning
	Architecture Definition	Is a visual representation of the above available and understandable? Are all interfaces with the stakeholders and concepts understandable? Is clear what does and does not belong to the project?	Is the project still acting within the boundaries of this contect? If changes are made, how do these affect this contextual layer? And what is the effect? Do you agree?	Can the PM prove that the interfaces work properly through the whole chain (to prevent claims)?		Strategic Management, Marketing, Public Affairs, Legal, Communication	Most likely a change has consequences at the lower levels and has impact on costs, time and risks. A contextual architecture should be drawn as a means to communicate to understand the functioning of the system to-be and what is in and outside scope. It should fit in the Enterprise Architecture or this one should change because	Contextual layer Business Architecture '- Information Architecture Application Architecture Infrastructure architecture
Enter-prise	Business and Mission Analysis	How does this project contribute to the strategy or is this project a consequence of a change in strategy (for public, change of law)?	How are all activitities in the project aligned to the (new) strategy?	Are KPI's in place to measure the outcome of the strategy and is the system providing the information for it?		Strategic and Tactical Management	When the strategy changes the project could encounter other circumstances than usual and has to adress it, also in the risklist	Business Architecture Business Use Case Model Project Plan / Risk list
	Stakeholders needs Requirements definition	Which internal stakeholders are involved in this change and in which role? Are internal SLA's needed?	Are all stakeholders involved according to their role? If not, who will solve this? Are the SLA's executable and according company's rules?	Do all stakeholders agree with the outcome, according to their role?		Tactical Management Supported by BPM and HRM Business Analysts	The project has to decide what within scope of the project and what not and has to be ready in due time to ensure the delivery of this project. When processes have to be redesigned tactical management have to decide with HRM if for instance the workers' council should be informed or consulted.	Corporate Strategy Business Architecture Business Process (Re-)design Requirements Definition
	Stakeholders needs Requirements definition	Which business functions will deliver the new or changed products and services? Are they already in place? Which business rules or regulations should be applied? Which risks have to be mitigated?	Are other products or services added? Why? Do you agree? How is the involvement of the stakeholders. How is the interaction between the business functions ensured?	Are all products and services agreed upon by each responsible stakeholder? Can their quality in use (as far as possible) be proved? Are KPI's in place on the interaction and does the system provide the information?	f g h j k l ;	Business Process Management Information Management Product Management Marketing, Public Affairs, Legal, Risk (security) Management Business Analysts	The interaction model between the business functions is part of the proces design: can the information be produced and gathered form all the busines functions at the intended time and place? The information model provides insight in the ownership of the data. Business Continuity, DPA en BIA are derived from these insights	Corporate Strategy Productmanagement (product and services) Business Processses Requirements definition
	Systems/Software requirements	Is clear to which extent the business functions will be(come) digital? Is Reuse before Buy before Build visible, if customised software is chosen, is the business case clear? Do processes have to change? Define which of the ISO25010 standards are most important for you and why.	Are the interfaces clear and working as expected (performance, business continuity (without producing unexpected customised software)?	Can they prove that these conditions are met and to which extent? Are processes and software aligned ?		Business Process Management Enterprise Architecture Solution Architecture Purchasing	Applications may force change of processes. Using customised software instead has to be decided within the constraints. Interaction between all systems must be possible, data only recorded once (or explained why not). The requirements on enterprise level are about interaction betweeen the business functions, and the constraints (ISO25010 and changing or not of processes). Purchasing can start Requests for Information	Corporate Strategy Enterprise Architecture Solution Architecture Business Process (Re-)design Requirements Definition Business Use Case Model
	Architecture Definition	Can a visual be made that shows which (intentional) software belongs to which business function and how these functions will interact internally and externally? Are alternatives shown?	Can it be shown in which areas progress is made and what will be delivered when? Can it be shown when ready?	Can they prove that the interfaces work properly not causing unexpected delays?		Enterprise architecture Solution architecture Purchasing Risk and Controlling	The solution has to fit within the enterprise architecture (or this one should change for good reasons): the goal and the current situation have to be compared to show some alternatives from which the client can choose. The order in which progress should e made depends on the interfaces and which process neeeds what to test	Enterprise Architecture Solution Architecture Requirements Definition Business Use Case Model Interface design
Enter-prise	Design Definition	Is a service and product design available? What is needed to create the new or changed products and services? Is this feasible? Are business functions to produce them present and ready? Does this require a change of the organisation model?	Are the products and services developed according the product design? Can (conceptual) products and services be delivered for marketing and pre-testing purposes to obtain feedback at an early stage?	According to their role, are all stakeholders happy with the products and services?		Board of Directors Tactical Management Planning & Control, Business Architecture Product Design Business Process Improvement	The products and services design should be valid, i.e. the data to produce these products must be available or made available during the project (if so, this must be added to the risk list) When business functions have to be created or changed, HRM can advise that the workers'council should be consulted or informed	Use Case Model UXP Design Process Design Requirements model
	Design Definition		Are al new concepts defined and agreed upon (and is the enterprise information model updated)?	Are the descriptions and definitions clear and unambiguous for all users?		Information Architecture GDPR-official Security architecture	At this level for some data it may be decided that they should not be shared and need a higher protection. For personal data are the GDPR-rules valid.	Information/Data Model Security architecture/Design Requirements model
People	Stakeholders needs Requirements definition	Do the new or changed business functions changes in jobs and is in- or decrease of staff required? Should training plans be made? Should the software be accessible? Which usability and quality in use aspects are important for the users?	Can, if needed, job descriptions be derived from the use cases? Could the software and prbe used for training and job descriptions to have skilled staff at the due date?	Is an adequate number of professional trained staff to operate the new system available? Are the forecasted number of resources met? Do people feel comfortable with the new system?		HRM & Tactical Management Training	The requirements for the workforce are conditions and requirements. The requirements of the people who use the system can be described as conditions derived form the ISO 25010 standards of usability and quality in use. Trianing plans have to be ready before going live: therefor using the software to deliver the training on he essential is wishful	Use Case Model UXP Design Process Design Requirements model
	Design Definition	Are specific target user groups involved and which goals have to be met? How do you want these user-groups to be involved?	Are users involved according to the plan? How's their opinion/emotion mesured during development and tests?	Are all user acceptance tests executed and are the requirements met? If not, do they prevent the transition?		Tactical Management (Supported by BPM and HRM) Marketing Product Management	Acceptance tests should be performed at all levels, strategic, tactical, operationla management and end users to ensure that cooperation is possible	Use Case Model UXP Design Process Design Requirements model
Usage Processes	Design Definition	Are there any specific requirements for the processes that have to be changed apart from those mentioned earlier? SHE? Sustainability? Error handling? Legal?	Is tactical and operational management involved to design and or verify the processes?	Is the functional suitability ensured? Does tactical and operational management understand and agree with the process design? Are the processes tested together with the new application(s)?		Business Process Modelling Control/Risk	the process design is necessary as far as needed for compliance and transfer of knowledge.	Use Case Model UXP Design Process Design Requirements model
IT	Stajeholders needs and requireents definition	Are there any specific requirements from the run or is governance-department, which could not be mentioned in the architecture?	How is ensured that the software is developed according to the standards of your organisation? Are the run-conditions met during development?	Is the system delivered according the standards and guidelines? Is error-handling delivered?		Operating Model & Governance, Security, Control	The IT standards should be delivered as requirements	
IT	Systems/Software requirements definition	Will the applications be developed and run by your own organisation or others? Is a tender necessary? When developed or run outside, are specific risk mitigation measures necessary? Which level of contracting is chosen? Are all requirements ready for the start? Are the product quality requirements (ISO25010) defined?	How is ensured that the software is developed according to the standards of your organisation? Are the run-conditions met during development?	Is the system delivered according the standards and guidelines? Is error-handling delivered?		Operating Model & Governance, Security, Control	The IT standards should be delivered as requirements	
	Architecture Definition	Is the Solution Architecture ready and reviewed?	is the solution architect esuring that the solution i developed according to he architecture?	Are the architectural requirements met?		Application Architecture Software architecture	Teh Solution Architect transfers the architecture to the Software Architect. They ensure that the soloution will be developed according the architecture	Solution Architecture Software architecture
	Design Definition	Can the requirements be mapped to the architectural visual representation? Is it made understandable which work packages contribute to which extent to the expected value?	Can they show that the derived specifications are validated by the users and their management	Is the system delivered according the standards and guidelines?		Analysts and developers	All requirements have to be organised in such a way that they can be understood by the teams, and duplication must be avoided	
	Systems Analysis	Which development path provides the most certainty that the efficient method of production is used in order to use the software as early as possible for marketing and training purposes and is deleverd on time?	Is the plan still delivering early value	Are working parts (tested) delivered according plan?		Planning & Control, Application Architecture	The software architect creates planning sequences and discusses these wit the project manager and the teams	
	implementation Integration	Is enough skilled staff available to perform the tasks? If not, are sourcing contracts availaible?	is progress according plan? Which impediments occur and can they be solved or is action needed?	How is visible that the software works according to teh requirements?		Project Manager HRM		
	Transition	Are there any specific measures needed for the transition? Fixed transition periods, down-period, or other)	Can the trasition be tested durng development?	Are all procedures ok?		TI Architecture		
IN USE	Benefits Tracking	Is the system delivering the intended value?		Is the system delivering the intended value?		Planning, Finance & Control		

	Verification	Systems test	Test Implementa	Working software	for the above and
Level	IS012207		METHOD	Kolom1	Input from/used by
Contextual	Business and Mission Analysis	Definition study Business Object Model Architecture Vision	Vision Document	Product owner	Marketing and/or Public Affairs/Legal Board of directors
	Stakeholders needs Requirements definition	Requirements model	Vision Document	Product owner	Marketing and/or Public Affairs/Legal Finance & Control
	Systems/Software requirements	Requirements model	Vision Document	Product owner	All stakeholders for this level
	Architecture Definition	Architecture Requirements	Software Architecture Document (SAD)	Product owner	All stakeholders for this level
Enterprise	Input upper level	Business Use Case Model Business Architecture	Business Use Case Model	Product owner	Corporate Strategy Productmanagement (product
	Design Definition	Use Case Model	Use Case Model	Product owner	Planning & Control, KPI's Business Process Redesign
	Design Definition	Information Model	Vision Document	Product owner	Quality, Security, Privacy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	Process Mangement
People	Stakeholders needs Requirements definition	Use Case Model	Use Case Model	Product owner	Business Process Redesign HRM Training Plans, UXP, Inclusion,
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	
Usage Processes	Stakeholders needs Requirements definition	Use Case Model Interaction Models	Use Case Model Supplementary specifications Interaction Models	User stories in Product Backlog	Business Process Modelling Control/Risk: security/error protection
	Design Definition	Use Case Model Interaction Models	Use Case Model Interaction Models	User stories in Product Backlog	HRM: job descriptions and training plans Security: authorisations Control: compliancy
	Verification validation	Acceptance criteria Use Cases	Acceptance criteria Use Cases	Demo, DoR and DoD	
Level	IS012207		METHOD	Kolom1	Input from/used by
Level	IS012207		METHOD	Kolom1	Input from/used by
		Waterfall	RUP	Scrum	
IT	Systems/Software definition	Use Case Model Interaction Models	Use Case Model	Backlog	Operating Model, Governance, Security and Maintenance
	Architecture Definition	Information and Technology architecture	Software Architecture Document (SAD)	User stories in Sprint Backlog	Operating Model, Governance, Security and Maintenance
	Design Definition	Use case specifications	Use case specifications + Additional requirement	User stories in Sprint Backlog	Sprints or build teams and maintenance
	Systems Analysis	Interaction Design and DB-design	Use case realisation, Class diagrams,	User stories in Sprint Backlog	for the above and Operations
	implementation Integration	Build, component or package	Build Implementation model	Working software	for the above and Operations
	Transition	Production Acceptance Test	Test Implementation Test Results	Working software	for the above and Operations
	Validation	User and Business Acceptance	Test Implementation Test Results	DoD	for the above and Operations

APPENDIX 5 - Projects

Appendix 5.1. PRIVATE

Appendix 5.2 PUBLIC

For confidentiality reasons submitted separately