

MASTER THESIS
INFORMATION SCIENCES



RADBOUD UNIVERSITY
FACULTY OF SCIENCE
INSTITUTE FOR COMPUTING AND INFORMATION SCIENCES

Software Development for the Public Sector

*The challenges that exist when developing software for the public
sector and how models can help to mitigate them*

Author:

P. Huberts
P.Huberts@student.ru.nl
s0814873

Supervisor:

Prof. Dr. Ir. Th.P. van der Weide
Th.P.vanderWeide@cs.ru.nl

Second Examiner:

Dr. P. van Bommel
P.vanBommel@cs.ru.nl

August 3, 2017

Abstract

The development of software for the public sector brings a number of challenges with it. Current development methods can handle some of these challenges, while some challenges cannot be handled or could be handled better. If errors occur during the development or implementation of a system the consequences could be severe, even to the point where the system will not be used in the end. The core of the problems is a lack of control over the development process. To solve this problem and the challenges that accompany it, an addition to current development methods is proposed. This addition shows how the use of models can help to structure the development process. This may increase the chance for successful implementation.

The interoperability model can be used to create the structure in this process. This model brings the organization, the development team and the infrastructure suppliers together. All these groups are required to ensure a successful implementation of the software system.

Preface

This thesis is the culmination of all the knowledge I gathered during my Master Information Sciences. It was the last thing that needed to be done before finishing my studies and becoming a university graduate. In the last six months I learned a lot about software development for the public sector. Thanks to my supervisor Theo van der Weide for helping me along the way, with new ideas or even simple tips on how the work could be done. Thanks to Patrick van Bommel for being my second examiner and reading my thesis. Thirdly, I want to thank Topicus for giving me the opportunity to use their contacts, in order to gather all the information I needed. Last but not least I would like to thank my friends and family for their support and motivation during the research, especially my girlfriend Annick.

Contents

1	Introduction	4
1.1	Problem Definition	4
1.2	Research Methodology	5
1.2.1	Development methods	5
1.2.2	Model based Development	6
1.2.3	Challenges in public sector	6
1.2.4	Improving the handling of challenges	6
1.2.5	Validation of development method	6
2	Software Development	8
2.1	Development Methods	8
2.1.1	Waterfall	8
2.1.2	Agile Unified Process	9
2.1.3	Scrum	10
2.1.4	Software development in Practice	12
2.2	Model Based Development	13
2.2.1	Model Based System Development	13
2.2.2	Object Role Modeling	14
2.2.3	Unified Modeling Language	14
3	Software Development for the Public Sector	16
3.1	Categories of Challenges	16
3.1.1	Organizational Power (OrP)	17
3.1.2	Politics (Pol)	17
3.1.3	Education (Edu)	17
3.1.4	Project Management Issues (PMI)	18
3.1.5	Ambiguous Business Needs and Unclear Vision (ABN)	18
3.1.6	Security and Privacy (Sec)	19
3.1.7	Finance (Fin)	19
3.1.8	Information Technology (IT)	19
3.2	Comparing Development Methods	19
3.2.1	Theoretical	20
3.2.2	Waterfall	20

3.2.3	Agile Unified Process (AUP)	22
3.2.4	Scrum	24
3.3	Practice	26
4	Improving Software Development for the Public Sector	29
4.1	Challenges not handled	29
4.1.1	Organizational Power	30
4.1.2	Security and Privacy	38
4.2	Challenges that could be handled better	39
4.2.1	Finance	40
4.2.2	Information Technology	41
5	Validation	44
5.1	Case Evaluation	44
5.2	Expert remarks	46
5.2.1	General Impressions	46
5.2.2	Points of Interest	47
5.2.3	Layers of Model	48
5.2.4	Coherence of Model	48
5.2.5	Language Based Approach	48
6	Conclusion and Discussion	49

Chapter 1

Introduction

1.1 Problem Definition

In the development of software for the public sector there are multiple stakeholders involved, each with different needs and values. Consensus about the problem definition (the problem the software is suppose to solve) is a prerequisite of getting consensus about the solution. You need to know what to problem is, in order to solve the problem[38].

Even if the problem is clear, the solution might not be. Some goals are hard to quantify. For example, it is hard to quantify what positive acceptance is and how it is a success factor in the development of an Information System. The answer to this question is different for every system. Getting positive acceptance from the public is not as straight forward as implementing a certain feature after which the system will receive positive support. It is not possible to know for certain what the public will see as positive, during the development process this could even change [18]. When there is news about a data leakage, security requirements can get more strict in order to convince the public of the safety of their personal data, so the answer of what positive support is can easily change.

This inherent complexity has many more causes. A more detailed explanation about the challenges in developing software for the public sector can be found in Chapter 3 on page 16. In 2013 an estimate of \$450 billion was spent worldwide on information technology. With a market this large, benefits of developing more efficiently could be enormous.

This thesis will look at how software development for the public sector can be further structured. The goal is to maximize the chances for successful software development. New possibilities for structuring can help the development process. A good structure will help organizations in the development process. It will enable the organization to overcome challenges.

The research question is: How can the use of models help structure the development of software for the public sector?

1.2 Research Methodology

The research is done in three phases. The first phase includes a literature study, the results are combined with information from practice. In the second phase the results are combined and a solution is presented. In the third phase, the solution is verified and validated.

1.2.1 Development methods

The literature study starts with a search for the three most used development methods. These three methods will be discussed in detail. A fourth method, derived from practice, is added. This method is in use by a software development team, that develops software for the public sector. This results in a total of four development methods.

The interviews are conducted with stakeholders and team members, from a project where software used in the national cancer screening program, is developed. Different stakeholders are interviewed to create a complete picture of the situation. This allows for multiple views on the same situation. To maximize the amount of information that is gathered, the interviews are semi-structured. This structure ensures the research is conducted in a sound scientific manner. At the same time it enables enough flexibility to gather as much information as possible.

The interviews will first be transcribed and then coded. The codes are created using a thematic approach. This method ensures a structured way of identifying challenges that have risen during the software development.

1.2.2 Model based Development

The next part of the literary study focuses on model based development. Research is done on how models can be used in software development and what some of the methods are on how to use these models.

1.2.3 Challenges in public sector

To answer the question of how to improve the software development process a few steps need to be taken. The first step is, clarifying the challenges when developing software for the public sector. This part of the study focuses on finding challenges in existing literature. The results will be a set of challenges that need to be handled. This could increase the chance for the successful development of software for the public sector.

The next phase will analyze each development method on how it handles the found challenges. All methods are systematically checked on if and how they handle each of the found challenges. This will result in a comparison between development methods. It will show what method can cope with these challenges most effectively.

1.2.4 Improving the handling of challenges

The challenges found in the previous phase that can be handled better are reviewed. The start of this review is why this method does not cope with this challenge. The results of this are converted into requirements for the solution. The requirements are the basis for the presented solution. The solution is designed to be added to the development method found in the previous phase.

To verify the proposed solution, arguments are presented why this solution is correct. At the same time it is argued why the solution will work when implemented in the development method.

1.2.5 Validation of development method

To assess the validity of this research, interviews are conducted. The aim is to check if the theoretical challenges, when developing software for the

public sector, exist in practice. At the same time this information is used to validate the suggested solution. It is assessed whether these will improve the software development process. The interviews in this phase are conducted with domain experts. The experience of the experts gathered during their careers is used to verify the quality of the solution. The results of this verification will be incorporated in the final method.

Chapter 2

Software Development

There are numerous approaches to software development. What approaches are used and how can they contribute to the successful development of a software system?

2.1 Development Methods

In this follow literary comparison the three most used software development methods are explained. The most used development methods are derived from research of Vijayasarathy and Butler(2016)[39]. According to their research Waterfall is used in 32% of the projects, Agile Unified Process in 28.1% of the projects and Scrum in 20.3% of the projects. Most of the development methods used are a combination of methods (45.3 %), forming a hybrid method[39]. The fourth described method, is derived from practice. It is used by a team that develops software for public sector.

2.1.1 Waterfall

The waterfall development method is the traditional way of developing software. This method can be compared to a real waterfall. You start at the top and work your way down to the bottom, as shown in Figure 2.1. After each step there is a review and verification. Only after this is done the project can move to the next step. It starts with the gathering of requirements, these are the needs of the customer. This is done by the

project manager before the start of the project. The goal is to understand user needs as best as possible[33]. After this is done, these requirements are used to create a design. This design includes an architecture of the software and the accompanying documentation. The third step is implementation, during this phase the actual software is developed. After the development process has finished it is time for verification. This is to ensure the developed software meets requirements regarding performance and quality. Once this is done the software is ready for release, this is the maintenance phase. It is important that software is maintained after release. This is done to ensure the software keeps functioning correctly[30].

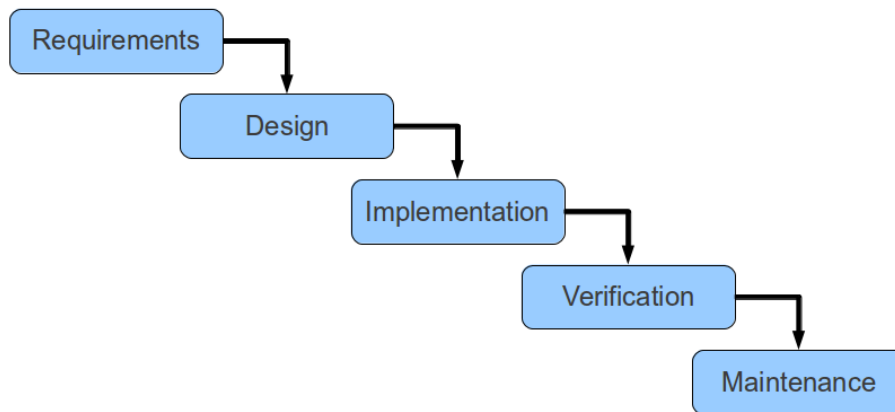


Figure 2.1: Waterfall Process[5]

2.1.2 Agile Unified Process

The Agile Unified Process (AUP) is a streamlined version of the Rational Unified Process. Development using this method consists of four main phases. Each phase consists of one or more iterations. The workload of different roles can be viewed in Figure 2.2. On the horizontal axes it shows what phase the project is and what the iteration of that phase (I1 is inception, iteration 1). On the vertical axes it shows the different roles. For each role, the height of the line shows the workload for that role.

The first phase is the inception. One of the goals of this phase is to create consensus among the stakeholders about the objectives of the project. At the same time funding for the project is secured [8]. The process of securing funding includes the creation of a schedule for the project. Also, the risks and feasibility are assessed in this phase.

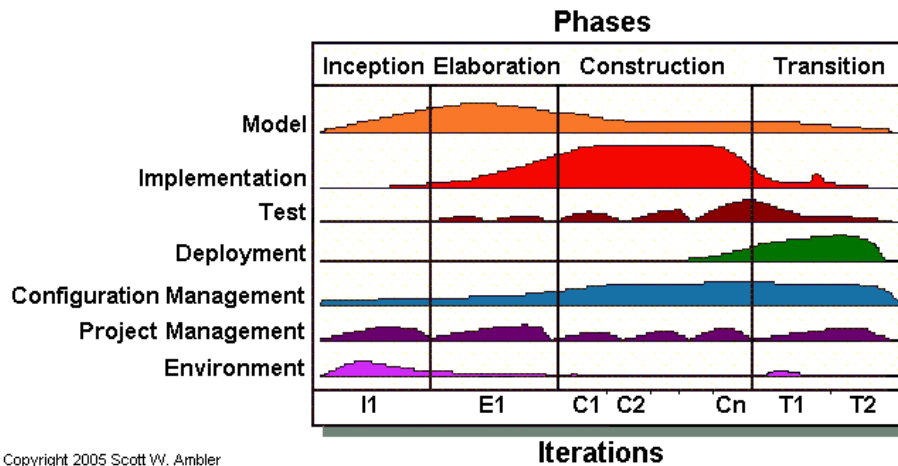


Figure 2.2: Agile Unified Process[7]

The second phase is elaboration. In this phase an architectural prototype is created. This prototype is created to identify the capabilities of the team. The architecture helps in further identifying risks of the project. It is important to note that the requirements that are used to create this prototype are not complete. They are only specific enough to create the architectural prototype.

During the third phase, the construction of the application is started. This phase can have multiple iterations. After each iteration it is assessed if the product is ready to move beyond preproduction. If this is not the case the decision is made if a new iteration is started or if the project will be canceled. Only when the product is ready to move beyond preproduction the next phase will started.

The last phase is the transition. This is when the software is extensively tested. If there are faults found, the stakeholders can accept these or can request a fix. When this phase is done the software is ready for release and daily use[7].

2.1.3 Scrum

The Scrum development method is an iterative process. Each iteration is called a sprint and ideally lasts circa two weeks. At the start of each sprint a list is made of which functionalities will be developed this sprint. The goal is to build the best software possible for the financial resources available. The

workflow of scrum is shown in Figure 2.3. The base of the scrum development method is the scrum team, this is the development team. Team members have a shared responsibility for the development process. The whole team is responsible for both success and failure.

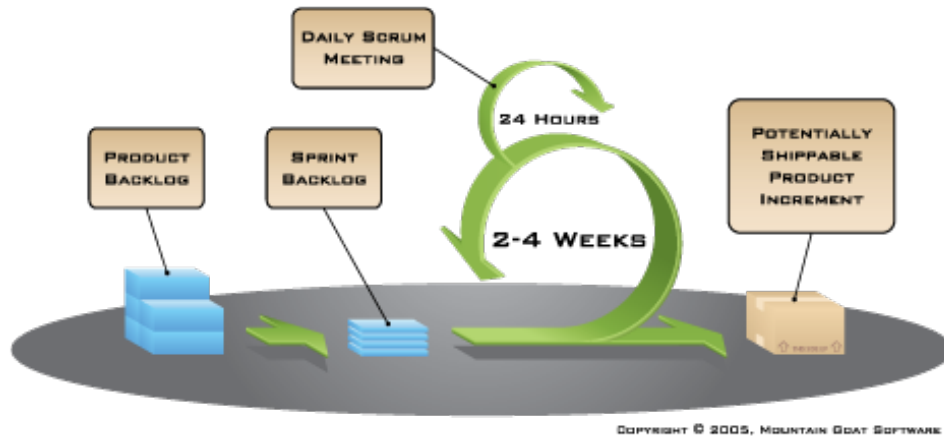


Figure 2.3: Scrum Development Process[3]

During the first phase of each iteration there is a meeting between the customer and development team. During this meeting requirements for this iteration are set. In practice this works effectively because customers know the features they need now. At the end of this iteration, the customer can use the features he needs the most. These features are delivered at the end of this iteration. At the same time, this meeting gives developers the possibility to ask questions about the requirements that need to be met [14].

The next phase is the actual sprint. This is where the features that were selected for this sprint are developed. There is a daily scrum meeting lasting at most 15 minutes. The attendance of customers in daily scrum meetings helps to keep them involved. At the same time it helps to reduce confusion. The whole team, developers and customers, know what is developed. This results in an attitude change from the customer: from ambivalence to becoming involved. However, it is important to note that the customer needs time to get familiar with this new role [16].

The last phase of the sprint is the sprint review. In order to get an idea of what went good and what could be done better, a meeting takes place. The purpose of this phase is to learn from each iteration and improve the overall process. Once this is done the process is adapted and a new sprint starts[26].

To ensure good communication between the customer and development team, a specialized role is created. This is the role of *Product Owner*. This role includes the responsibility to represent the needs and ideas of all stakeholders. The goal of this role is to create a single point of contact between stakeholders and developers. The *Product Owner* is part of the development team and does not work for the customer. It can be challenging for the *Product Owner* to identify and involve all stakeholders. Without the identification of the stakeholders it is impossible for the *Product Owner* to do his job effectively[25].

2.1.4 Software development in Practice

In practice the development process is different from theory. Even before the actual development starts, a tender is written. At this point it is decided what the requirements for the project are. During the next phase companies can enter a bid for this tender. Each of these companies study the tender and write a response on if and how they can meet the requirements. With this list a quote of the costs is created. This part of the development process is very similar to waterfall development. It takes advantage of benefits of waterfall to create a long term planning and setting long term goals. If the company wins the tender, it moves to the next phase. This is where the software will be developed.

When the development starts, the method changes to Scrum. This change is made to take advantages that iterative development brings. This flexibility is used to handle change requests during the development process. To ensure that the developed feature is the correct one, a lead from the business is involved. For every feature that needs be developed the following path followed. It starts with a kick-off this is when the developer talks with the product owner, lead and a tester about what exactly needs to be developed. The lead is an end-user who works for the costumer, who has the same rights and privileges as the product owner. At the end the developer knows what he needs to do. The next stage is the part where the new feature is developed. In the third phase the same participants as in the kick-off give their opinion on the results of the development. The last stage is testing. To confirm the feature works it is tested by a team from the customer. This ensures that the features that have been developed reflect the requirements set by the customer. Should the developed feature require any change, this can be done without any problem.

2.2 Model Based Development

Model-driven approaches can be used to add an abstract layer to the development process. Models are build to better understand the software that is being developed. At the same time it often shows opportunities to reuse and simplify code [11]. Other purposes for using models include[17] [29]:

- Understanding existing code
- Maintain consistency throughout the application
- Clarification during ad-hoc meetings
- Designing & Refactoring
- Design review
- Explanation to customers
- Documentation
- Simulate usage and performance

It is important to take into account that a model does not have to be complete to be useful. The goal of using a model is to simplify a problem, not to have a complete model. Modeling parts of the application can even be an opportunity. For example, security can be left out to simplify the model. This simplification helps in getting an overview on that part of the application [27]. At the same time this simplification can be used to make the development more understandable for the customer. The models described in this section are examples of how models can be used to improve the development process. There are a number of tools available that allow fast and easy creation of models. However, the general thought still remains that these tools are not available[12]. A Possible method for using models in development is Model Based System Development.

2.2.1 Model Based System Development

Developing a system using Model Based System Development happens in four stages [34][29]. These stages are:

Functional Model Abstraction of the system functions that need to be delivered. This model is used to simulate and evaluate the application before it is build.

Architecture Allocated Functional Model In this model, parts of the modeled functions are allocated to architectural elements. This includes both the software and hardware architectures.

Virtual Prototype The real-world system is represented virtually. Such a prototype can be used by developers to test and optimize strategies for handling communication faults in the system. This is includes all communication to, from and within the system.

Architectural Models The architectural model does not address the system's functional details. It addresses the following two models:
Software Model: this is model about the tasks and their relationships.
Hardware Model: this model is composed of the computational nodes and interconnection network.

2.2.2 Object Role Modeling

Object Role Modeling(ORM) is a method to model and query an information system at the conceptual level. ORM models the world in terms of objects that have roles. An added benefit of ORM is the ability to use natural language[22].

ORM focuses on the data model, this is more stable than for example models that focus on the functions used. ORM is not only a graphical language but also a textual language. In a graphical language it is easy to express common constraints. It however is not always possible to express them clearly and in depth. The use of text can then be used to explain constraints in more detail. The graphical structure allows for direct translation to textual definition [23].

2.2.3 Unified Modeling Language

The Unified Modeling Language(UML) is used to model numerous systems. This includes anything from an enterprise information systems to web applications. Some of the purposes for using UML are:

Language for Visualizing Developers can create a model to visualize ideas they have. Visualizing a model using UML gives the ability to create

uniformity among models. This enables different developers to create models which can be used by the entire team. At the same time it ensures information is written down and stored for use in the future.

Language for Specifying The goal of specifying is to create a model that is complete, precise and ambiguous. UML allows for this usage in all important stages, analyzing, designing and developing.

Language for Constructing It is impossible to program an application using UML. The model can be used to generate code for an application, which can then be used to finish the program. It is also possible to generate a model from existing code.

Language for Documenting All the different purposes of UML make it perfect for documenting an application. This includes documenting models like the architecture, designs, prototypes and releases. This ensures that at a later stage clear documents are available [11] [37].

Chapter 3

Software Development for the Public Sector

The generic challenges surrounding software development are well-known. An example of this is, project management issues. Besides these challenges there are additional challenges when developing for the public sector. For example, even after decades of developing software for the public sector it is still hard to estimate the size of the project. Even experienced managers estimate the amount of work wrong in more than 50% of the cases [20]. At the same time it is not possible to have a development method that works for all projects. Different development methods are suited for different project sizes. In this Chapter the following question will be answered: What makes software development for the public sector more difficult?

First it will be explained what type of challenges exist when developing software for the public sector. When this is clear, a comparison is made with the four development methods discussed in Chapter 2.

3.1 Categories of Challenges

Anathopoulos et al.(2016)[9] conducted a literature review about why e-Government projects fail. In this review they found eight main factors. These will be explained, in order to gain insight into the challenges that exist when developing software for the public sector.

3.1.1 Organizational Power (OrP)

Most organizations in the public sector have a large number of employees. The size of an organization influences how a new system can be introduced. The introduction of a new system influences the workflow. Changes to this workflow can lead to resistance towards the change[13]. Typically a larger organization will encounter more resistance to change. It is important that the organizational structure can be used to overcome these obstacles. Beer(2000) talks about different ways to dissolve this resistance and create support for the change[10].

Besides the internal challenges, external challenges are also a factor. There are many stakeholders involved in the development of software. These have different needs and different backgrounds. Consequently, the requirements could become more complicated. Especially when the needs of one stakeholders group are the opposite of another group[18].

3.1.2 Politics (Pol)

Developing software for the public sector includes the challenge that politics impose short-term goals on the software development process. This is a consequence of the political system in use. Periodically there are elections, that can result in a shift in the political system. For example in the Netherlands this is at least every four years. After the elections, a new government is formed. This could have different priorities compared to the previous. This results in a planning that is stable for the current election period. This reoccurring change in leadership make it that long-term planning is difficult. A new government can change the planning of the previous government[40].

3.1.3 Education (Edu)

Managers in general lack specialized knowledge. This is the knowledge that professionals have and need to do their jobs. Usually managers do not need this knowledge to excel at their jobs. There are cases in which this knowledge is needed. Evaluating a new software product is one of those cases.

A lack of knowledge also occurs with end users. In this case this could both have to do with reluctance and/or the inability to adopt and adept to the new technology [20]. The lack of knowledge could also be a factor that

can cause implementation failure of a new system. At the same time it is important that the users know the value of a new system. This includes both the citizens and the staff. Users must be qualified to operate the new system. Either by training them or creating an intuitive system. So it is important that each group of users has the knowledge needed to operate the new system[6].

3.1.4 Project Management Issues (PMI)

There are number of sources that can cause project management issues. These are the four most common causes[9]:

Underestimation of timeline. It is important that progress of the project is known, how much work is done. Not having an accurate system in place to monitor the timeline, can result in a project that is not finished on time.

Weak definitions of requirements and scope. It is important to have a clear definition of requirements and the scope of the project. When it is not clear what has to be made, the project can not deliver high quality software. If the requirements or scope are not clear, the project manager should take action to ensure they are clarified. Good development starts with clear objectives[19].

Inefficient risk analysis and management. Projects can encounter setbacks. The manager of a project can ensure that there are systems in place to handle these setbacks. When this is done inefficiently, not only are not all risks known, there is also no plan on how to deal with these risks. This could result in extra costs, delays or even failure of the entire project.

Unsuccessful monitoring and measurement. Having access to monitor and measurement tools, can help a manager in determining how the team performs. When the performance drops, this could be an indication of problems. Without accurate monitoring it is hard to spot problems in time, when there is still time to fix them[24].

3.1.5 Ambiguous Business Needs and Unclear Vision (ABN)

Projects in the public sector can suffer from intangible benefits. These subjective benefits cannot be quantified. An example of this is: improving

the quality of life. What quality of life is, is different depending who is asked. The strong opinion of popular figures can even change public opinion. Thus these requirements can change easily. Resulting in extra costs and/or longer development time [31]. In other cases there are multiple stakeholders each with a different views on the subject. This could cause a situation where it is impossible to reach a consensus regarding goals and objectives [21].

3.1.6 Security and Privacy (Sec)

It is important that applications keep the information they store safe. Without trust, citizens will not use these services when they ask for their personal data [6].

3.1.7 Finance (Fin)

In a report from the United States accountability office from 2012, 9% of all software development projects are completed within budget and delivered on time. This has partially to do with problems during the analysis of what needs to be done. In some of these cases it was not completely clear what the impact of a new system is. Without this information it is impossible to estimate the cost of the project[36].

3.1.8 Information Technology (IT)

The problems that grouped under this section are related to infrastructure, data, compatibility and information management. In a more practical sense this means that successful implementation can be inhibited by procuring a poor product and insufficient long term planning [35].

3.2 Comparing Development Methods

In this section the development methods in chapter 2 are compared to the challenges shown in the previous section. In Table 3.1 the results of this comparison are visualized. Each method will receive a ✓ when it is able to handle the challenge, a ± when it is able to handle to challenge to some degree or a ✗ when the development method is not able to handle the challenge.

This method of comparison will give the ability to directly compare the different software development methods.

	OrP	Pol	Edu	PMI	ABN	Sec	Fin	IT
Waterfall	✗	✗	✗	✓	±	✗	✗	±
AUP	✗	✗	✓	✓	±	✗	±	✓
Scrum	✗	✓	✗	±	±	✗	✓	✗
Practical	✗	✓	✓	✓	✓	✗	±	±

Table 3.1: Method comparison

3.2.1 Theoretical

Three theoretical methods are compared. These are chosen based on research from Vijayasathy[39]. The methods in this comparison are: Waterfall, Scrum and Agile Unified Process.

3.2.2 Waterfall

The choice to develop using the waterfall methods is often made, when developing large software systems. The choice to use the waterfall approach is made to benefit from the extensive requirements and design phases. But can the benefits of the Waterfall software development handle the challenges that come with developing software for the public sector?

Organizational Power Before the start of the project, the product managers tries to understand the user’s requirements. After this period there is no further communication between the organization and the development team. The review and verification of this period needs to be of high quality. This will ensure that there is no gap between the business and the software that is in development. Without active communication between the software development team and the business it is impossible to know how the organizational power is divided. The development team does not know specific requirements per stakeholder. Only generalized knowledge is present. Thus the waterfall development method is unable to handle this challenge.

Politics This development process has an extensive requirements and design phase. Requirements set at the start of the project are used during the whole project. It is not possible to make changes during the project, without making it a new project. One of the main challenges from Politics is the lack of long term goals, they can change with each new government. Thus the requirements for the project can change. Therefore, the waterfall method does not handle this challenge well.

Education Before starting the requirements phase the project manager tries to understand the user's needs. This is done by communication with the end users of the new product. The manager uses this time to fill the gap in specialized knowledge that the product manager has. However, there is nothing in place to ensure the right knowledge is passed down to the end users. The waterfall method is unable to handle this challenge.

Project Management Issues At the time the project is started the whole scope is thought out. The goal is to have a complete picture of everything that needs to be done. This should minimize the possibility to have any project management issues. All issues that could rise should have already been thought over. Either beforehand or during one of reviews at the end of each phase. So this challenge can be handled by waterfall.

Ambiguous Business Needs and Unclear Vision The goals of the project are set at the beginning, this should imply that the goals are clear. However, in the public sector the goals can be intangible. If the public opinion changes, this can result in the changing of one of the goals of the project. Thus as long as the goals of the project do not change waterfall ensures that there is a clear vision and clear business needs. If the goals change both the vision and business needs will be unclear. Thus this method can handle the challenges only if the goals do not change.

Security and Privacy Requirements regarding security and privacy can be part of the scope. When this is part of the requirements it will be part of the design and then it will be implemented. However, this is not mandatory. It relies on the project manager if and to what extent these requirements are implemented. As long as this is done the model can handle security and privacy. As waterfall does only handle security and privacy when it is added to the requirements, this challenges is not necessarily handled.

Finance Waterfall defines the whole project at the start. Thus it is hard to envision what problems need to be overcome. Without an effective method to identify the impact on the new system, it is almost impossible to have control over the costs. The lack of financial control shows this model is not able to handle this challenge.

Information Technology The second phase of the waterfall development method is used to create a design of the entire system. At this point the whole project is thought out. This includes everything that has to do with the infrastructure needed. However, when changes are required, there is no way in waterfall to do this in the current project. This results in a method that can handle this challenge as long as requirements do not change. So the method can handle it to some degree.

3.2.3 Agile Unified Process (AUP)

The Agile Unified Process is suited for all project sizes. The roles that are needed for the development are added when needed. This ensures that the knowledge that is needed at that point is available. It is possible that one person fulfills more than one role.

Organizational Power The first phase of the Agile Unified Process tries to create consensus among stakeholders about the objectives of the project. This is the only time before the construction, the opinion of all the stakeholders is sought. Even with the agreement about the goals, it is not specified how they should be reached. It is important to agree on how a goal is reached. One way could lead to more resistance to change, than others. As organizational resistance is still a risk, this challenge is not nullified.

Politics At the start of the project the goals are set. These are used until the end of the project. This development method cannot handle the changing of the goals. These are the general objectives that need to be reached. The Agile Unified Process is able to handle smaller changes. This would be changes limited to how a goal is reached, not what goal is reached. Every phase can consist of multiple iterations. At the start of each iteration such a change could be added. If a change takes place because of a shift in politics, this can most likely not be handled because these types of changes are usually extensive. The political shift causes changes about what goal needs to be reached.

Education At the start of the process all stakeholders get involved. At this point they can all voice their opinion. This allows all stakeholders and thus experts to fill a gap in knowledge if there is one. At the end all stakeholders are involved to find faults and give feedback on the application. This can result in a request for a fix, but does not specify what the fix is. It is possible that the fix is education for the end users. The workflow of the Agile Unified Process helps to void this challenge.

Project Management Issues Agile Unified process is split into different phases, only in the first phase there is an effort to create consensus. This effort only focuses on the main goals. Consensus is the base of the project. When there is a chance for conflict at a later stage the project manager can refer back to this consensus. This results in the power a project manager needs to mitigate this challenge.

Ambiguous Business Needs and Unclear Vision The guidelines of the Agile Unified Process set understanding the business process as one of the goals of the modeling phase. The model helps to create an abstract representation of the real world. Identifying business needs is done in a structured manner. However, the Agile Unified Process does not specify how this can or needs to be done. So this challenge is partially nullified.

Security and Privacy It is possible to incorporate security and privacy in the scope of the project. If the stakeholders reach a consensus that it is one of the goals, this will be implemented. This is not an effective way of handling this challenges. It is possible that stakeholders do not see the urgency of this challenge and therefore do not give it appropriate priority. Without any obligation this challenge cannot be handled.

Finance During the first phase of the project an estimate for the costs is made. However, this estimate is never reviewed or adapted during the lifetime of the project. So the start is good, but the follow up is lacking. Which is, keeping this estimate up to date, is described as a tip but nothing more. It is realistic to expect some sort of change during the project lifetime. This is why the challenge is partially mitigated.

Information Technology During the second phase of the Agile Unified Process an architectural model is created. The fact that an architectural

model exists in the handling of this challenge. It gives the ability to handle any questions about information technology that rise.

3.2.4 Scrum

The scrum development method is suited for small to medium sized projects. The average team size is seven people plus or minus three people. When developing larger systems the decision could be made to include more people but this would endanger parts of the scrum development method. For example the daily stand-up should last at most 15 minutes. With a larger team each member has less time to speak.

Organizational Power The product owner handles the communication between the organization and the development team. The decision in which order the work is done, is made by the product owner. His decision is leading. When developing software for the public sector it can be hard to identify all stakeholders and the power they have. There is no mechanism in the development method to take this power into account. Thus it not possible for the product owner to do his work. Resulting in a challenge that cannot be overcome.

Politics Scrum is a method to develop software in a iterative manner. This gives the possibility to cope with changes to the requirements of the project. Thus when in politics the goals of the project are changed, these can be implemented starting in the next sprint. It is no problem that the long term goals can change. The only consequence could be that it will take some time to adjust the project to the new requirements. This has to do with the size of the changes needed to handle the new requirements. So Scrum can handle this challenge.

Education According to the scrum development method the product owner has all the necessary knowledge to make the correct decisions about the project. This challenge states that managers in general lack the specialized knowledge the end users have. This gap in knowledge could cause problems. It is not an option to make multiple persons product owner. The specification of this role states its a single person. The lack of this specialized information can for example cause that changes to the workflow are overlooked. This will cause problems later in the process. There is also no way to communicate to the end users what choices have

been made and how the software is best used. Without the knowledge needed it is almost impossible to use the software as effective as it could be. Scrum does not handle this challenge well.

Project Management Issues The product owner determines what work is most important. Based on this, he selects the work that needs to be done in the next iteration. According to the theory, the product owner knows everything from the business. He has to represent every involved stakeholder. The variety of stakeholders make this almost impossible. Their needs can not only differ, but they can conflict. This development method does not help the product owner in resolving these situations. It depends on the product owner if this challenge can be handled. Thus this challenge can be handled when the product owner excels at his job, thus when he knows everything about the business.

Ambiguous Business Needs and Unclear Vision The product owner should ensure that the business needs and the vision are both clear. The goals should be reflected by the backlog (the To-do list). The items on the backlog are prioritized by importance. This single point of communication between the business and the developers, creates a single point of failure. Some of the goals that need to be reached can be hard to quantify. In those situations it could be beneficial to include both developers and people from the business to come to a solution. This is why scrum can handle this challenge in some situations.

Security and Privacy All the work that needs to be done is added to the backlog. It is possible to add requirements that ensure the security and privacy. The product owner determines if items get added to the backlog. Next, the product owner determines when these will be developed. So the scrum method can incorporate features that ensure security and privacy. However, this is all without obligation. So the scrum development method does not take security and privacy into account.

Finance At the start of each sprint it is clear what has to be done during this iteration. A development team can do a certain amount of work for a certain cost. At the end of each sprint a new working version of the software is delivered. This ensures that the process can stay within budget. There is always a working version of the software that can be delivered. The goal

of the scrum process is to create the best software for the money available. Thus scrum can handle this challenge well.

Information Technology The information technology has no specific place in the scrum development process. Scrum is a methodology for developing the software not to manage a project where software is developed. Thus scrum does not take this into account.

3.3 Practice

There are numerous companies developing software for the public sector. Some of this is delivered on time and within budget. So how does a software company tackle these problems? In this section it is explained how this company tackles these challenges. Further details on how this team works can be found in Chapter 2.1.4.

Organizational Power To help the customer in gaining organizational power, the development teams acts as a facilitator. The goal is to create consensus on how and why choices are made. This helps to overcome resistance that the customer faces in implementing a new system. Resistance can even be external. A stakeholder from an external organization confirms this. He said, *If we get access to the system, our work would be much easier.* The goal of this stakeholder is to get access the new system. His organization is an example of stakeholders that are not directly involved but can cause obstacles. The success of creating consensus lies with the expertise within the development team. The success of handling this challenge depends on a few individuals. As the handling of this challenge is not part of the development method, it is not mitigated.

Politics At the start of the project it is clear what the goal in general is. The details need to be worked out, but this can easily be handled by the iterative development process. Before the development starts a tender is written. Different companies place a bid for this tender. Eventually one of the companies wins and gets to develop the new system. The tender results in an agreement between the company and the organization that wants the new system. One of the stakeholders said, *in the end the minister makes the decision on whether the software will be developed.* Even with this agreement it is possible that a new minister has a different opinion than

its predecessor. Thus this can lead to a request to change or cancel the agreement. Even in the unlikely event this will happen, this will not be a problem for the development team. As the team switches to an iterative development method, the ability to handle changes is added. The team can handle any changes that are requested as a consequence of political change.

Education To ensure knowledge from the end users is available during the development, end users are added to the team as needed. A user is added based on the specialized knowledge needed for the development of that part of an application. Part of the development process is to hand off the new developed software to the end users. This is done in different ways. Either by organizing a training session or by the delivery of user manuals. So this challenge is mitigated.

Project Management Issues According to the definition of scrum the product owner is part of the development team and not working for the customer. In this practical approach the choice has been made to make an employee of the customer, product owner. This with the added role of a lead for each issue, should ensure that every stakeholder gets heard. This close cooperation with the customer is in place to minimize project management issues. This ensures that this challenge has almost no influence on the success of the project.

Ambiguous Business Needs and Unclear Vision If the software development team faces ambiguous business needs or an unclear vision, it will go actively look for ways to find answers to the questions that have risen. In practice this involves finding someone who is allowed to make a decision about the question at hand. A member of the development team, *We try and involve as much people as we can, to map the requirements out as best as possible.* At the same time, by involving end users in the development process, ambiguous business needs are minimized. All the measures taken by the development team result in clear a vision and clear business needs. So this challenge can be handled.

Security and Privacy The security and privacy requirements are based on the rules and regulations that are in effect. In practice the development team ensures that there are no problems with this challenge exist. One of the consultant on the project said, *privacy and security are very important, we have a dedicated lawyer who says what laws we need to abide by.* So there

is a system in place to know what requirements need to be set regarding security and privacy. However, there is no specific part of the development method that ensures the correct handling of this challenge. Thus it cannot be handled.

Finance The software development process starts with a tender. The result of this tender is a contract between a public organization and a software development company. In this contract it is specified what software is going to be developed and what the payment will be. As long as everything that is being developed is specified in the contract the costs can be perfectly controlled. One of the consultants said, *There are discussions, whether or not a feature is part of the original agreement.* It happens that an extra feature that was not part of the original contract needs to be developed. The development of extra features results in an increase in costs. This part of the development process is hard to control. So this method is able to handle this challenge as long as signed contract includes everything that needs to be developed.

Information Technology At the start of the project it is clear what Information Technology is going to be used. The specifics of what kind of technology are based on the requirements from the software. If these requirements change, the technology used can change. However, there is no architectural model created of the system. The room for improvement results in that this challenge is partially solved.

Chapter 4

Improving Software Development for the Public Sector

As described in Chapter 2, each software development method has its strengths and weaknesses. The theoretical methods are designed to be used in generic software development projects. The practical method is tailored to work when developing software for the public sector. When developing software for the public sector the challenges as described in Chapter 3 need to be overcome. This is shown in the comparison of how each method handles the challenges. The practical method is best able to cope with these challenges. It is able to handle most challenges. However, two challenges cannot be handled and two challenges that leave room for improvement. For each challenge a possible solution is suggested.

4.1 Challenges not handled

There are two challenges that the practical method cannot handle. The first is the Organizational Power. This is about who has what to say within the organization. It is important that an organization is ready for a new software system. The second is Privacy and Security. When the product is in use both privacy and security must be ensured. If the data in the system is not safe or could be extracted or tampered with, the new system will not be used.

4.1.1 Organizational Power

Why can the practical method not handle the Organizational Power challenge?

The implementation of a new software system will require the organization to change. It is impossible to replace a software system without changing the business process that uses the software. If the new system would not improve the business process there is no reason to replace the systems that are already in place. The practical development method depends on the experience of the team members, to help overcome any resistance within the organization. If for some reason these people are indisposed, this could become an obstacle for the success of the project.

What is needed so this method can handle it?

A structured method needs to be in place, that is able to handle resistance within the organization. This could be done by using a model. As explained by Dutton and colleagues[15], numbers, charts and showing the impact on the organization, will help legitimizing a viewpoint. A model can use this legitimization to create readiness for the change. The model will use this legitimization in creating readiness for the change. When the resistance is overcome, the development team will be able to move the process forward.

How can this be done?

Usually the requirements are set using natural language. This is done in order to give stakeholders the ability to verify them. It could occur that requirements are vague. In those cases it is especially important that requirements can be verified. Using natural language when setting requirements brings its own set of challenges. For example, if natural language is used, it is impossible to know for sure all participants will use the same semantics for the used sentences. Besides lacking a formal syntax and semantics, natural languages suffers from a variety of problems. A natural language suffers from inherent complexity. For most individuals it takes years to learn a language. At the same time words can have multiple definitions and different groups have different styles in using the language. In many cases there are multiple interpretations for the same sentence. Such misinterpretations can result in the development of the wrong requirement.

This can be solved by using a formal language. Such a language has a formal definition of both its syntax and semantics[32]. However, a formal language is difficult to use in practice. A semi-natural language combines the best of both worlds, having an unambiguous language that is easy to use. By setting the right restrictions to a natural language, it is transformed into a semi-natural language. As each domain-language is slightly different from the others, the restrictions are specific for each domain. Using unambiguous languages, gives the ability to accurately translate them from one domain to another. People with different linguistic and cultural backgrounds can improve their ability to communicate effectively. There are multiple examples where semi-natural languages are used. One of the best known examples is the language used in the aviation sector. Their semi-natural language lets pilots communicate with air traffic controllers without miscommunication. Another example is communication during military operations. Miscommunication can directly cause the loss of lives, this must be avoided at all costs. By using a semi-natural language every sentence over the radio has exactly one meaning. This decreases the chance of unclear communication.

The groups involved in the development can be split into multiple levels of refinement. For example, the domain language used in the organization differs from the domain language used by the software developers. Therefore we can assume each level has its own specific domain language. So we can identify the language L_i as the language at level i and the semi-natural language on the same level as D_i . The languages on different levels are related to each other. The languages need to be related to allow for the required inter-level communication (the communication between different levels). As the languages are related, there is a possibility for translation. The use of semi-natural languages allows for the creation of translation schemes between the different levels. These schemes are used to create a perfect translation between different languages.

The Interoperability Model When this perfect translation of the languages is possible, different stakeholders can effectively communicate. Requirements can be effectively set. Every stakeholder will understand what the other stakeholders are saying. More specifically, miscommunication is not a possibility. In most cases requirements relate to a set of demands the new application must meet. This is not enough for successful implementation. To systematically create all requirements for a successful implementation, the change is split into five layers of interoperability(the ability to exchange information and work together)[4]. All layers are equally important. As flaws on one layer can cause problems in the success of the complete

application, communication between the layers is a must. The requirements on one layer influences the requirements on the other layers. A layer can only communicate with the layer directly above or directly beneath it. This is done to ensure a correct translation between the different layers. If information needs to be passed between layers that are not adjacent, this should go through the layers that are in between. This ensures that all the information is translated step by step. Correct communication ensures that the requirements complement each other and are not conflicting. Besides the five layers there are two columns spanning all layers. All these layers and the two columns combined give the basis for a successful implementation. Figure 4.1 is a visual representation of this model.

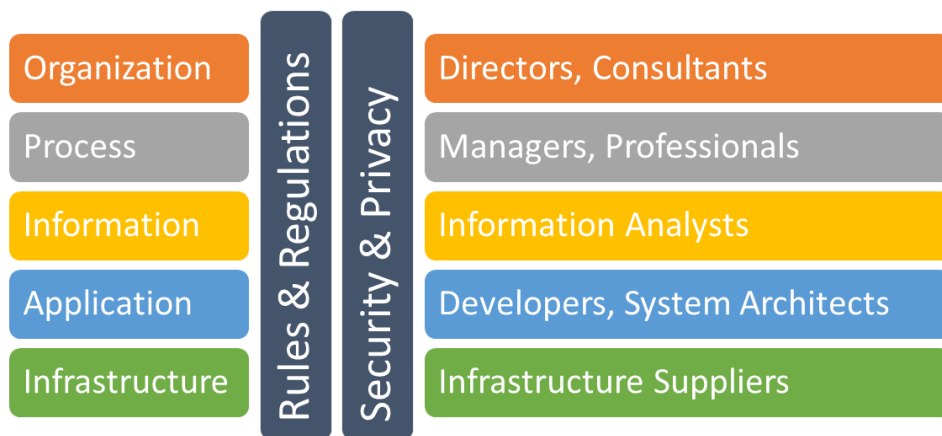


Figure 4.1: Interoperability Model [4]

- **Organization Layer** The implementation of a application can change the way the organization functions. At the board level decisions have to be made who is involved and how responsibilities and privileges are divided.
- **Process Layer** A new or changing system will have consequences for the business processes. By identifying the required change, the processes can be adjusted in order to create organizational readiness for the implementation of this system. This layer is used to specify how the work is going to be done using the new system. The requirements for this layer are set together with the managers and professionals of the organization.
- **Information Layer** In a software system, information is stored and processed, this is described in the information layer. An agreement is made between professionals and information specialists. The result of these requirements is that it is known what information is needed and what information needs to be stored.

- **Application Layer** All relevant information for the application is gathered by developers and system architects. This information is gathered from the professionals and application managers and in some cases suppliers. The result of this is all the information needed by the development team to develop the system that is needed by the professionals.
- **Infrastructure Layer** The new software system will have specific needs of how the technical infrastructure needs to be configured. The infrastructure suppliers make arrangements with the developers and system architects.

The interoperability model is structured like this to ensure a logical order exists. This order is comparable to the structure of other models, start with the generic and work to the specific. The decisions made on the organizational level about who is involved and how responsibilities and decisions are divided, are similar for most projects. The second layer is more specific, it is about how this change will effect the business process. Going down to the third layer, it discusses what information is needed by the professionals. It also includes what information the application needs to meet the requirements from the professionals. The fourth layer is a more specific version of the layers above. It holds the requirements for the software system. This includes how the application will function and what functionalities must be available. The fifth layer is about the technical needs the application has. These requirements are the most specific, the systems needs this kind of hardware.

There are two columns that span all layers. The obligations set by these columns need to be taken into account when setting the requirements for each layer. The columns are conditions for the process. As these conditions can span multiple layers, the two columns are created.

- **Security and Privacy Column** The Security and Privacy need to be included in the requirements on all layers. The combination of the requirements on all these layers can ensure the system is secure and the privacy is ensured. To enable the setting of the correct requirements it must be clear what conditions surrounding security and privacy must be met. These conditions are set in this column.
- **Rules and Regulations Column** The rules and regulations that are applicable to the entire process are converted into requirements and gathered in this column. When setting the requirements of each layer, these should be used as a starting point.

An argument could be made that these columns could also be handled by the different layers. However, this would open the possibility that these subjects are underestimated in the other layers. As a result, the needed requirements will not be set in every layer. Another reason why the columns should be separate and spanning all layers is that in order to ensure requirements are met, they must be set on all layers. For example when security measures are only taken on the application layer, these measures could become futile when an employee exports the data to a thumb drive and loses the drive. So even with all measures taken on one layer, they become futile by not setting them on another layer. In this example the breach could be prevented by requirements set in the organizational and process layers. Such a requirement could be a ban on the use of external storage devices.

The combination of a semi-natural language and a model that will ensure that no stakeholders is forgotten, is the basis for successful implementation of a new system. The inclusion of every part of the company in the creation of the requirements for a new system will help in overcoming the resistance to change, as described by Beer and Nohria in their Theory O[10].

Where To start There is no specific layer that has to be used as a starting point when using this model. There are two scenarios possible when wanting to use this model. In the first scenario, there is a single natural starting point. In the second scenario, there are multiple natural starting points.

When there is one natural starting point, this should be chosen as such. When this naturally happens, it is possible to set at least some of the requirements for the layer in question. These requirements could then be used when communicating with the other layers. This results in the setting of requirements layer by layer.

It could be possible that there are multiple logical starting points. In that case, the first thing that needs to be accomplished is that if there is a gap between the natural starting points, these are filled in as soon as possible. The connecting of these layers will ensure that the requirements that are set complement each other. This increases the chance for a successful implementation.

Usage Strategy After a starting point is chosen, the requirements for the layer in question need to be set as described in the definition of that layer. During the process of settings these requirements, it is important

to search for rules and regulations that apply and what requirements need to be set regarding security and privacy. During the process of setting the requirements it is important to ensure that, & regulations and security & privacy are consulted, these also need to be updated if not complete. Therefore, it is possible that the requirement used as a starting point need to be adjusted to fit within the requirements set by the rules and regulations and the security and privacy columns.

When the requirements have been set for the current layer, one of the adjacent layers is chosen and the requirements for that layer are set. During this process there is communication between adjacent layers. This is done to ensure that the requirements set on the different layers consolidate each other and no conflicts occur. The setting of requirements for a new layer is done for each of the five layers. At this point is it clear what the requirements are for each part of the implementation of the new system. This will maximize the changes for a successful implementation.

Correctness of Model The goal of most IT systems is to improve the quality or efficiency of the work done in an organization. When implementing a new IT system this will change how the organization functions. To increase the chance of a successful change process in an organization there is a need to manage this change. It is important to take the opinion of employees into account and create readiness within the organization for this change[10]. Otherwise resistance to change could cause difficulties during the implementation of this new system. Therefore, it is important to take the organization into account when creating a IT system.

On the organizational level decisions have to made about two subjects. How the organization is affected by the change and where responsibilities lie in the new situation. This creates a basis for a successful adoption of the change within the organization. If the information on the process layer is already available. These requirements are analyzed. Using the results it is determined what the consequences for the organization are. To ensure a successful implementation the organization must be able to handle these consequences. Thus, requirements must be set in such a way, that the organization is supported to function as best as possible.

The process layer describes how the organization does its business. This work is done by the professionals and managers of the organization. A new application will influence the workflow of the organization. If these changes are not considered it will be impossible to develop a system that will benefit the organization. The goal of this layer is, to use the knowledge that exists

within the organization. This is used to make an informed decision on what requirements should be set. When the process layer is used as a starting point, the professionals and managers in the organization are contacted. They have the knowledge needed to set the requirements. Doing work day in day out gives them the knowledge on what is good and what can be improved. If the information in the organizational layer is available, this is used as a basis for the process layer. There is communication between the organizational layer and the process layer. The division of responsibilities and privileges has direct influence on the process. Persons or groups identified in the organizational layer must be considered when setting the requirements in the process layer. If this is not done, a mismatch could occur causing problems at a later stage. If the requirements in the information layer are already set, the requirements of this layer are used to create a first outline for the new business process. It is known what information will be available. Thus, a decision can be made on how this information will be used in the new process. Again, this is done together with the professionals and managers. They will have detailed knowledge of the process. This expertise can help in finding challenges in the implementation of the new process.

In the information layer, it is described how the data is stored and processed. This layer could be used as a starting point when using the interoperability model. Information specialists and professionals from the organization discuss what information could be useful for them and why it is useful. This results in all information that is needed and why it is needed. When it is known what information is used, a translation can be made into how the information needs to be processed. The what, why and how information is used, combined, results in the requirements for the information layer. The information layer is present between the process and application layers to make the transition clearer. How is information used and processed, is important in both the process layer as in the application layer. Without this layer, the translation would become much harder and require interpretation by the stakeholders involved. As previously discussed, the necessity for interpretation of information, should be avoided. Consistency between layers and stakeholders is paramount. If available, information about how data is used in the process layer is further specified in the information layer. This is where additional requirements are set on how information will be stored and processed. When the application layer is available, the information that is stored will be described in this layer. This gives an overview of what information is available and gives the possibility to add new requirements to this dataset. At the same time, it is described how the information needs to be processed, this will result in a set of requirements. To ensure that all the possibilities of interactions between the different layers are correct, they are all analyzed.

The application layer specifies the requirements for the application. This can be used as a starting point when using this model. When this layer is used as a starting point there is an idea for a new application. To set the correct requirements for this idea the developers and system architects are involved, that are eventually going to change the idea into an application or new feature. In order to get to know the idea from top to bottom. When this is done they involve professionals and managers, they will eventually use the new idea. At the same time the infrastructure suppliers are contacted. These two groups are involved to ensure the idea will work in practice, both in the business environment as on the infrastructure. When the requirements set in the information layer are available. These are used to effectively and accurately specify how the application needs to be configured. To ensure this is also compatible with the business processes stakeholders from the company are included in this process. When the infrastructure layer is available the requirements set in this layer are used as a basis for the requirements set in the application layer. The choice for a certain infrastructure can cause some technologies not to work as efficiently as they could. In those cases, the requirements in the infrastructure layer directly influence the requirements in the application layer.

The infrastructure layer is used to specify what technical infrastructure is needed. Suppliers and developers set requirements on what is needed. This results in conditions needed to run the application. This layer can be used as a starting point. If this is done, the infrastructure suppliers plan with the developers and system architects on what requirements should be set to ensure a correct functioning application. If the application layer is already available. This is done in almost the same manner. Additionally, the specific requirements from the application layer must be considered.

As shown above it is possible to start with the model on any layer. After a starting point is chosen, the other layers are included step by step. This is done in a structured manner to ensure consistency across the different layers. Including the organization in the software development helps the development process. It ensures that the organization is ready for the new system. This will improve the quality of the software and efficiency of the process. This model takes two aspects into account. The first is the development process and the second the organization. Both are important when setting the requirements. To avoid conflicting requirements, there is contact between the layers. When a conflict is found between requirements, changes are made to solve them. During this contact a decision is made what requirement is more important. This can result in the changes of requirements on a previous layer. In the end there will be consistency among all requirements. Contact between all layers is needed to ensure the correct implementation of rules & regulations and privacy & security. It is

impossible to create requirements on one layer and satisfy all requirements set in the columns. However, a successful implementation of the system depends on it meeting these requirements.

4.1.2 Security and Privacy

Why can the practical method not handle the Security and Privacy challenge?

Ensuring security and privacy is up to par, depends on the individual developer. The developer takes security and privacy into account when developing a new part of the software. There is no part of the development method that is specifically focused on ensuring the security of the system and the safety of the data inside the system.

What is needed so this method can handle it?

As the software is being developed for the public sector, it becomes subject to the laws that are set to ensure the safety of these systems. For a system in this sector it is paramount that a structured method for verification is available. This structured process guarantees the security and privacy of the data in the system. Such a structured method will safeguard against mistakes that could impact the security or privacy. At the same time it allows for an accurate impact assessment of how a new feature impacts the security and privacy.

How can this be done?

One of the columns of the interoperability model is Security and Privacy. In order to create requirements for that system ensure the security and privacy. A conceptual language needs to be used. It is paramount that every requirement needs to be explicit and not open to interpretation. A manner to create the requirements in a effective way is part of the problem. The starting point of ensuring the system is safe, is that it is clear how this needs to be done. Good and clear requirements will help in the development process.

Different models can help in visualizing different parts of the total

implementation. Data models can help in identifying what parts of the application contain sensitive data and how that data is connected to other parts of the application.

Figure 4.2, is an example of an Object Role Model. The structure of the data can be explained to anyone. By choosing a model that has a description of the data in it (Example: Athlete(.nr) was born in State) even stakeholders that have no training in the usage of this type of model can understand it's basic meaning. As a result, these stakeholders can engage in a discussion where such a model is used. For example in Figure 4.2, if there is a discussion about athletes and the state they are born in, in relation the the impact on security and privacy. Such a model ensures that nothing is overlooked and no part of the application is forgotten. Otherwise it would be easy to forget that the states are also coupled to a sport with a rank. Thus an adjustment to the state an Athlete is born in could have consequences to how the ranking of a state in a sport works.

The use of a model to identify threats regarding to privacy and security, can be helpful. It is a tool simplify complex situations The use of such a model helps in better understanding a complex situation [28].

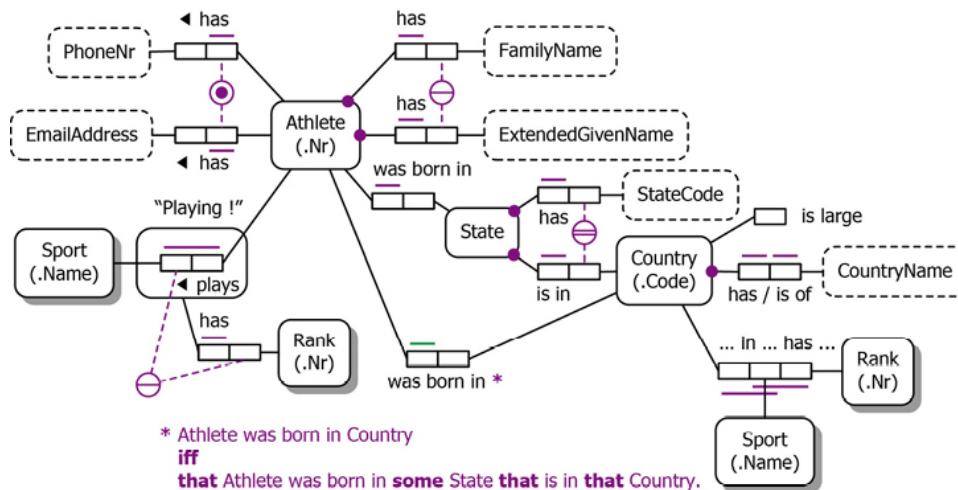


Figure 4.2: Object Role Model[22]

4.2 Challenges that could be handled better

Two of the challenges can be handled by the practical method to some degree. The first is Finance. Developing software for the public sector

involves a tender and thus a contract. In this contract the costs are specified. However, developing additional features will cost extra. The second is Information Technology. It is not described how the infrastructure is taken into account when developing software.

4.2.1 Finance

Why can the practical method not handle the Financial challenge?

During the development process it is possible for stakeholders to make requests for features. While agile is suited to handle these requests, it is difficult to estimate the exact size of requests. The requirements that are preset can be handled without any problem. There needs to be additional possibilities to handle requirements that change.

What is needed so this method can handle it?

If there are tools available that help in determining how much work a new feature or change request brings, this would help the development team. It enables the team to quickly and accurately respond to such a request with details on how much effort it will cost to incorporate the request. With the ability to accurately determine the size of such a request, it is better possible to estimate the costs.

How can this be done?

To estimate the size of a change and the corresponding costs, it is important that it is known what exactly needs to change. To ensure the request is correctly understood a semi-natural language needs to be used. The goal is to set the requirements without any ambiguity. Taking the requested change and using the interoperability model to identify the changes needed on every layer gives a complete overview. At this point this overview needs to be further specified. This results in the size of the change per layer. Thus resulting in a good way of identifying different part of the application that will be effected by this change.

For example, in Figure 4.3, a class diagram of an order processing system is depicted. The overview shows off what parts the program is made up. For

every part it shows what functions and what variables that part uses. When estimating the size of an alteration or addition to the program such a model could be very useful. For example, if there is a request to add the ability to have a different shipping address. This model helps to show where changes need to be made. In this example the information needs to be added to the customer and to the order. The models helps in identifying where changes need to happen. In this case it shows that the address is not only stored in the Order-class but in the Customer-class as well. The principle is the same for more complex situations. The model helps in the identification of what parts of the application are linked.

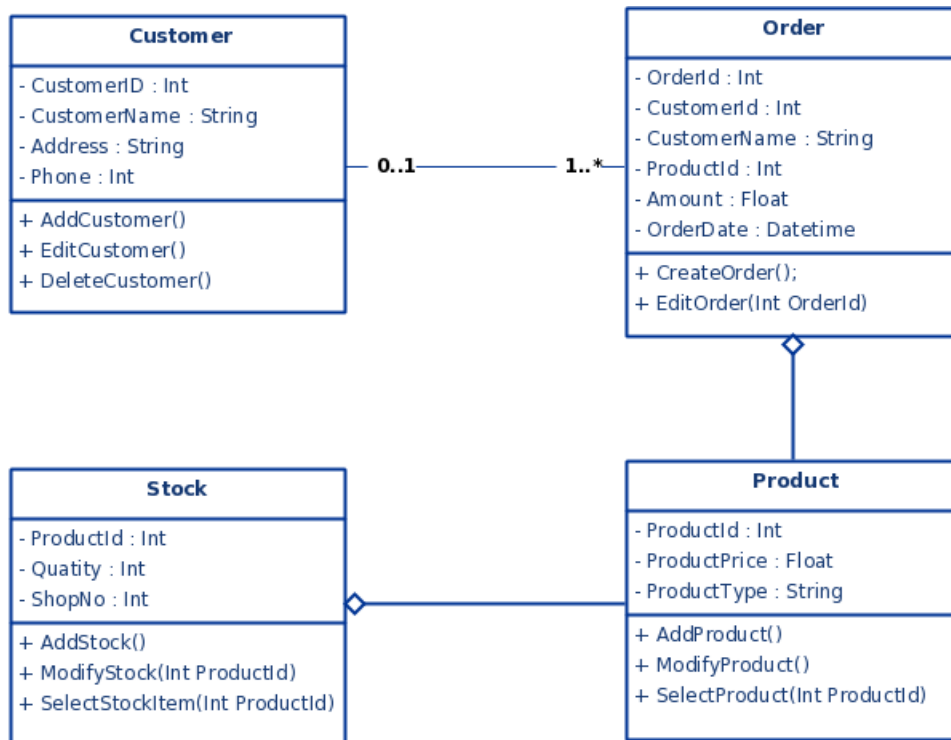


Figure 4.3: UML: Class Diagram for Order Processing System [2]

4.2.2 Information Technology

Why can the practical method not handle the IT challenge?

Scrum is a method for developing software systems. The challenges that come with how such a system is implemented on different types of infrastructure is not part of this process. Developing a system is not the same as implementing that system.

What is needed so this method can handle it?

An overview gives the ability to discuss problems related to the data, infrastructure, compatibility and information management in a structured manner. A model ensures that it is clear to everyone involved what parts of the application depend on which parts of the infrastructure.

How can this be done?

Using the interoperability model as a starting point, this challenge is part of different layers. The requirements for the infrastructure and compatibility are set in the infrastructure layer. The data and information management are defined in the information layer. The use of the interoperability model ensures that the communication and the description of requirements is not only understood by everyone but also is unambiguous.

Setting requirements on the information layer could be done by using a model of the information system. There are different modeling languages that have the ability to model the infrastructure of an application. One example of this is Archimate. Figure 4.4 shows a model of an insurance company. The yellow section is about the business, the blue section is about the application and the green section is about the infrastructure. This model gives extra information about how the infrastructure is set up. Only looking at the application layer it seems that there are two application (CRM system and Financial application). The infrastructure layer shows that even though these are two different applications, they run on the same application server. Using such a model ensures that no part of the implementation is forgotten, there is a good product to deploy and it is known what the planning for this deployment is.

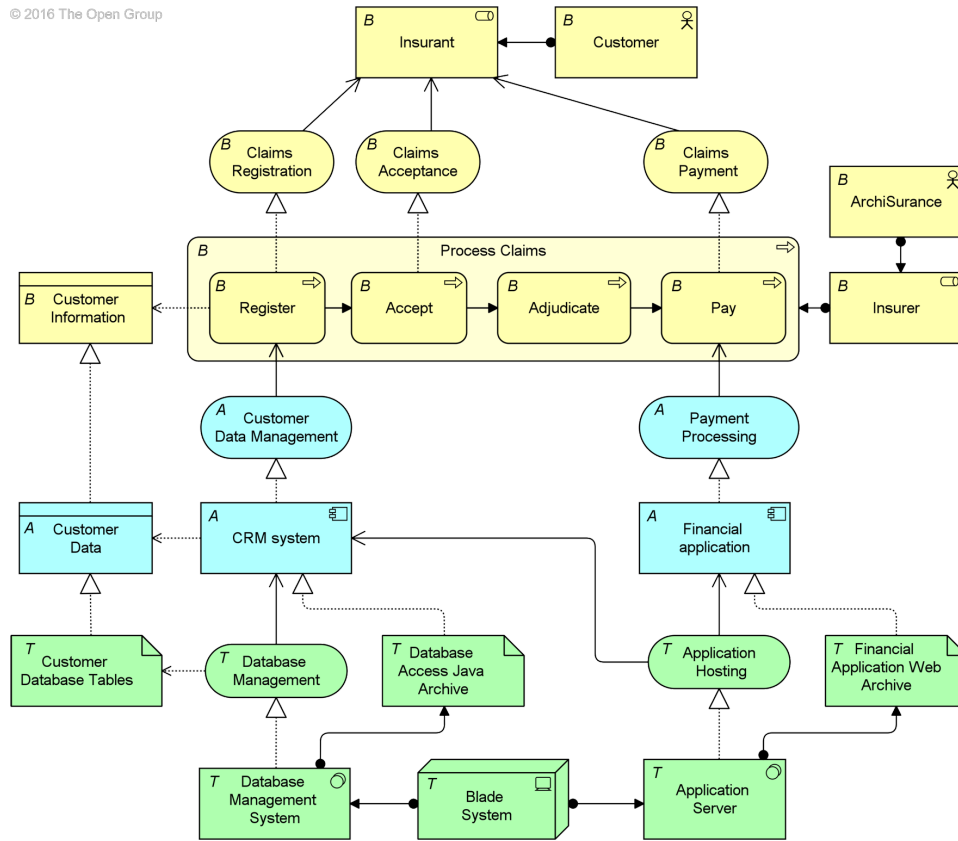


Figure 4.4: Archimate model [1]

Chapter 5

Validation

In this chapter the question 'is the correct problem solved?' will be answered. It is possible that a solution is found for a problem that does not exist. This chapter will explain if the proposed solution is valid for the described problem.

5.1 Case Evaluation

In this evaluation it is identified if the challenges that are found in the literature are also exist in practice. Interviews were conducted to gather information about the software development process in practice. This source is also used to gather information about if this development team encounters these challenges. It is possible to encounter these challenges, but that their method for development is able to cope with these challenges. This information is gather from the six interviews that were conducted for this study.

Organizational Power During the development there needs to be a deliberation with different stakeholders. Sometimes different stakeholders have opinions that matter. This is an example of the organizational power challenge. The manager from the development team said, *We have only one costumer, but there are multiple parties who are involved in the decision making* Every choice in such a case will lead to resistance and this needs to be overcome. The development team tries to fill this gap by taking the lead in this conversation, because they want to keep moving forward in the

development process. As said by the manager of the development team, *We have the experience in process chain automation and the customer lacks this knowledge. This is why we sometimes take charge in this process.* This will result in the initiator not having the power they need to insure this deliberation happens effectively.

Politics It happens that the software development team has to deal with politics. These changes are outside the field of influence of the development team. The team just has to adapt to these changes as it is almost impossible to change them.

Education A lack of knowledge with the individual that needs to convey the requirements from the professionals to the software developers, opens the possibility that the requirements are not set correctly. This will result in the development of the wrong new feature. To fill this gap the development team includes users in the development process. The result of this is that the product owner is not the single point of communication between the development team and the organization the software is being developed for.

When a new system is being developed, this will result in the changing of the workflow. A different workflow needs to be explained to the professionals within the organization and there needs to be training for these professionals to allow them to work with the system correctly. In order to mitigate this challenge, training for personnel is organized. However, it still happens that this new workflow is not followed, this results in errors.

In both cases the lack of knowledge results in the challenges during the development or implementation that cannot be overcome and thus will result in the failure of the system.

Project Management Issues Project management issues occur in every project. There is close contact between the development team and customer, so the team is able to handle most issues on its own.

Ambiguous Business Needs and Unclear Vision During the software development process one of the first steps is to schedule a kickoff for the issue in question. The goal of this meeting is to clarify any questions about what exactly needs to be developed. This part of the process is included to minimize situations where the vision or business needs are not clear.

The development team adapted its process to ensure this challenge can be handled.

Security and Privacy The application that is in development needs to meet rules and regulations regarding security and privacy. The presence of this requirement directly results in the existence of this challenge.

Finance At the start of the project an agreement between the software developer and the public organization has been signed. In this agreement it is stated what is going to be build and how much it will cost. Even with this agreement the financial challenge exists as new features are added, or original features are changed. Every time a part of the application is build that is not part of the original agreement, this will lead to extra costs.

Information Technology The knowledge of hardware is present with the developers. In the cases where this knowledge lacks, the department that manages the infrastructure is involved. This department has specific knowledge on what the consequences of certain requirements are and what needs to be done in order to make sure the performance of the application is up to par.

5.2 Expert remarks

The solution is presented to two experts that have worked in software development for at least 10 years. The goal is to verify the theoretical results and see if they can be used in practice. The remarks are collected separately to minimize the chance for bias, in interview results.

5.2.1 General Impressions

The organizational power challenges, can cause major challenges. During the development there is not one organization that has be taken into account but multiple organizations. This increases the challenges surrounding organizational power as there is not one person that has authority in all organizations. When working with public organizations it is not as simple as to develop software, as it would be for the commercial sector.

In a commercial setting there is always a person responsible that has the power to make decisions. In the public sector this is not the case. In each organization there are people that have power and their power is considered legit, even if it is technically not. Thus each stakeholder has to be taken into account when making decisions.

As the interoperability model finds its origins as a communications model, it is able to handle multiple responsible persons per layer. The goal of each layer is to set requirements and thus to set who is responsible. If this is not one but multiple persons this can be handled.

The inclusion of the organization in the software development process was thought to be a good solution. Usually the implementation of a new software system is not thought of as an organizational change. The experts both thought this was in fact the case. Getting the organization to the point where this conclusion is shared, is helpful. This shared conclusion can be the basis of an implementation plan. Numerous theories about how organizational change works, can be used in this plan.

5.2.2 Points of Interest

The first expert endorses that the flexibility of the interoperability model with the inherent usefulness of models can help in the successful development of software for the public sector. One note the first expert gave was that although the use of models is useful, it is hard to choose the right model. Each model has its strengths and weaknesses, he would like to have suggestions on what models to use to set requirements per layer (future research could take this into account).

The second expert would like to have more specific suggestions on how the model could be used in collaboration with existing models. It is also important, what to do when requirements do not have to be set on all layers. For example, it is possible that a change on the infrastructure has no influence on the application and thus not on any of the above layers. This is an addition to how the model should be used. When a change has no influence on some of the layers these can be excluded in the setting of the requirements. The decision of if they have influence, needs to be made by the ones responsible for that layer. For example, in the application layer, the developer can assess if a change has influence.

5.2.3 Layers of Model

The experts both thought the layered approach is effective. It enables to select parts of the model that are useful at that point. The layers in the model help in effectively steering a conversation. This is useful to ensure decisions are made. It is easy for a complex conversation to get off-topic. For example, when talking about how the process works, it is not important what infrastructure is needed. At that point decisions about the process need to be made.

5.2.4 Coherence of Model

The step by step usage of the model was thought be a good approach. The experts agreed that it ensures that every part of the implementation is discussed. When developing for the public sector there are other suppliers that have to be taken into account. By not only having the customer and developers in the model but other suppliers as well, this makes the model complete. These are all factors that influence the successful implementation of a software system.

5.2.5 Language Based Approach

The employees from the customer that are included, are not trained in the use of models. Most models have a certain level of abstraction. By choosing an language based approach, the ability to explain subjects that are unclear, is gained. This works both ways, from developers to employees and vice versa. The experts found this low-threshold very important.

Chapter 6

Conclusion and Discussion

There are numerous development methods available, some more used than others. When comparing these theoretical methods with a method used in practice, the practical method is a combination of theoretical methods. The development process in practice adapts to what is useful and what works. While this research only shows the method that is used by one software development team, this conclusion is in line with the research from Vijayasathy[39]. However, the research from Vijayasathy was not specific for software development for the public sector. To verify if these generic results are applicable when developing software for the public sector, further research is needed to verify this.

The theoretical challenges that exist when developing software for the public sector were confirmed by the experts. It should be further specified that these challenges are not necessarily unique to the public sector. Some of these challenges are found in all types of development processes. The eight challenges described cause the most failures in developing software. Therefore it is important to be aware of them, in order to minimize the impact of these challenges.

In order to overcome these challenges the interoperability model is introduced. It ensures the needs of the business and the infrastructure suppliers are taken into account when developing software. The goal is to have clear communication. It is important for the development team to know what the needs of the business are and what infrastructure is available. To set the correct requirements in a certain layer, other models (such as UML, ORM, Archimate, etc) can be used to help clarify the requirements that need to be set. This research showed an example that some of these

models could be used to help set the requirements in a layer. Further research could be conducted what models could be best used for certain layers. If this results in a few recommendations of what model to use, this would increase the chance of this model being used in practice.

Implementing a new system into an organization, changes the processes within that organization. If a business process changes, this is organizational change. The implementation of a new system, should be treated as such. At this point methods describing how to change an organization can be used to facilitate the change. The users of the system must want the change, otherwise they will resist it. This resistance could lead to a failure in implementation. More specifically it enables the users of the interoperability model to steer the conversation and to identify what subjects need to be discussed. While the model is not specific on how to give substance to each layer, this is where its power lies. It can be adjusted to solve the challenge at hand.

The interoperability model can be added to almost all development methods. These methods generally describe in what order work needs to be done and what the goals of each phase are. They do not go into detail on how the work should be done. The use of the interoperability model is an example of how work can be done. Adding this model to the development method will help the development team. It will enable the team to keep structure during the requirements phase of the project/cycle. This ensures that the organization and infrastructure suppliers are taken into account. Thus maximizing the changes for the successful development of the system.

Bibliografy

- [1] ArchiMate® 3.0 Specification. <http://pubs.opengroup.org/architecture/archimate3-doc/toc.html>.
- [2] Class Diagram UML Order Processing.
- [3] Scrum Development method.
- [4] Wat is interoperabiliteit? <https://www.nictiz.nl/standaardisatie/interoperabiliteitsmodel>.
- [5] Waterfall development method.
- [6] Tamara Almarabeh and Amer AbuAli. A general framework for e-government: Definition maturity challenges, opportunities, and success. *European Journal of Scientific Research*, 39(1):29–42, 2010.
- [7] Scott W. Ambler. The Agile Unified Process (AUP) Home Page. <http://www.ambysoft.com/unifiedprocess/agileUP.html>.
- [8] Scott W. Ambler. Feature Driven Development (FDD) and Agile Modeling. <http://agilemodeling.com/essays/fdd.htm>, 2014.
- [9] Leonidas Anthopoulos, Christopher G. Reddick, Irene Giannakidou, and Nikolaos Mavridis. Why e-government projects fail? An analysis of the Healthcare.gov website. *Government Information Quarterly*, 33(1):161–173, January 2016.
- [10] Michael Beer and Nitin Nohria. *Breaking the Code of Change*. Harvard Business Press, 2000.
- [11] Grady Booch. *The Unified Modeling Language User Guide*. Pearson Education India, 2005.
- [12] Mauro Cherubini, Gina Venolia, Rob DeLine, and Andrew J. Ko. Let’s Go to the Whiteboard: How and Why Software Developers Use Drawings. In *Proceedings of the SIGCHI Conference on Human Factors*

in Computing Systems, CHI '07, pages 557–566, New York, NY, USA, 2007. ACM.

- [13] Antonio Cordella and Niccolò Tempini. E-government and organizational change: Reappraising the role of ICT and bureaucracy in public service delivery. *Government Information Quarterly*, 32(3):279–286, July 2015.
- [14] Nagy Ramadan Darwish and Salwa Megahed. Requirements Engineering in Scrum Framework. *Requirements Engineering*, 149(8), 2016.
- [15] Jane E. Dutton, Susan J. Ashford, Regina M. O’Neill, and Katherine A. Lawrence. Moves that matter: Issue selling and organizational change. *Academy of Management Journal*, 44(4):716–736, 2001.
- [16] Tore Dybå and Torgeir Dingsøy. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10):833–859, August 2008.
- [17] Jens Eickhoff, Albert Falke, and Hans-Peter Röser. Model-based design and verification—State of the art from Galileo constellation down to small university satellites. *Acta Astronautica*, 61(1–6):383–390, June 2007.
- [18] Marcelo Carvalho Fernandes, Antonio Juarez Alencar, Eber Assis Schmitz, Monica Ferreira da Silva, and Rigel Procópio Fernandes. Acknowledging the Effect of the Depreciation of Tangible and Intangible Benefits upon the Evaluation of E - Gov Projects. <https://pdfs.semanticscholar.org/42d3/6bf786247559ab29d0259687ade470037c0a.pdf>.
- [19] J. Ramón Gil-García and Theresa A. Pardo. E-government success factors: Mapping practical tools to theoretical foundations. *Government Information Quarterly*, 22:187–216, 2005.
- [20] Shaun Goldfinch. Pessimism, Computer Failure, and Information Systems Development in the Public Sector. *Public Administration Review*, 67(5):917–929, September 2007.
- [21] Joydeep Guha and Bhaskar Chakrabarti. Making e-government work: Adopting the network approach. *Government Information Quarterly*, 31(2):327–336, April 2014.
- [22] Terry Halpin. Object-role modeling (ORM/NIAM). *Handbook on architectures of information systems*, pages 81–103, 2006.

- [23] Terry Halpin and Anthony Bloesch. Data modeling in UML and ORM: A comparison. *Journal of Database Management (JDM)*, 10(4):4–13, 1999.
- [24] Leon A. Kappelman, Robert McKeeman, and Lixuan Zhang. EARLY WARNING SIGNS OF IT PROJECT FAILURE: THE DOMINANT DOZEN.
- [25] Marta Larusdottir, Aasa Cajander, Peggy Gregory, Gilbert Cockton, Dina Salah, Kati Kuusinen, and Gerolf Nauwerck. Stakeholder Involvement in Agile Software Development. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction, NordiCHI '16*, pages 139:1–139:3, New York, NY, USA, 2016. ACM.
- [26] M. Mahalakshmi and M. Sundararajan. Traditional SDLC Vs Scrum Methodology—A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6):192–196, 2013.
- [27] Stephen J. Mellor, Tony Clark, and Takao Futagami. Model-driven development: Guest editors' introduction. *IEEE software*, 20(5):14–18, 2003.
- [28] Suvda Myagmar, Adam J. Lee, and William Yurcik. Threat modeling as a basis for security requirements. In *Symposium on Requirements Engineering for Information Security (SREIS)*, volume 2005, pages 1–8, 2005.
- [29] R. Nossal and R. Lang. Model-based system development. *IEEE Micro*, 22(4):56–63, July 2002.
- [30] Kai Petersen, Claes Wohlin, and Dejan Baca. The waterfall model in large-scale development. In *International Conference on Product-Focused Software Process Improvement*, pages 386–400. Springer, 2009.
- [31] Marta Raus, Jianwei Liu, and Alexander Kipp. Evaluating IT innovations in a business-to-government context: A framework and its applications. *Government Information Quarterly*, 27(2):122–133, March 2010.
- [32] Keijo Ruohonen. *Formal Languages*. 2009.
- [33] Vicki Sauter. The Traditional Waterfall Approach. <http://www.ums1.edu/~hugheyd/is6840/waterfall.html>.
- [34] S. Sendall and W. Kozaczynski. Model transformation: The heart and soul of model-driven software development. *IEEE Software*, 20(5):42–45, September 2003.

- [35] Judith Sligo, Robin Gauld, Vaughan Roberts, and Luis Villa. A literature review for large-scale health information system project planning, implementation and evaluation. *International Journal of Medical Informatics*, 97:86–97, January 2017.
- [36] The Tercio Pacitti Institute, Federal University of Rio de Janeiro, Brazil., Marcelo Carvalho Fernandes, Antonio Juarez Alencar, Eber Assis Schmitz, Monica Ferreira da Silva, and Petros Sotirios Stefanias. Evaluation of Agile Software Projects in the Public Sector: A Literature Review. *Journal of Software*, 11(3):312–325, 2016.
- [37] Hendrik Jan van Randen. *Inleiding UML: analyse & ontwerp van software*. Pearson, 2013. Google-Books-ID: s02EmgEACAAJ.
- [38] Anne F. G. van Woezik, Louise M. A. Braakman-Jansen, Olga Kulyk, Liseth Siemons, and Julia E. W. C. van Gemert-Pijnen. Tackling wicked problems in infection prevention and control: A guideline for co-creation with stakeholders. *Antimicrobial Resistance & Infection Control*, 5:20, 2016.
- [39] L. R. Vijayarathy and C. W. Butler. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software*, 33(5):86–94, September 2016.
- [40] Teresa Waring and Philip Maddocks. Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future. *International Journal of Information Management*, 25(5):411–428, October 2005.