COSMETIC STORE MANAGEMENT

SALESFORCE NAAN MUDHALVAN

PROJECT REPORT

Submitted By

SRI DURGANANDHINI G(611420104078)

VANITHA R (611420104089)

VIDHYA S (611420104093)

VARSHA S (611420104090)

in partial fulfilment for the award of the

degree of

BACHELOR OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING

MAHENDRA ENGINEERING
COLLEGE FOR WOMEN

TIRUCHENGODE,NAMAKKAL-

637205

2.7.6 Kanban

Kanban is a method used to improve the way a company can work and how they can develop further to be more efficient and therefor create a greater business value. The method puts a lot of effort to adjust itself into the organizations primary goals and to be ready for rapid change. A major aspect within Kanban is the usage of a delivery flow system which provides knowledge and limits the amount of *work in progress* by using visuals. These visuals prevents too much of a workload to be put on the workers, and also prevents the system from accepting to little work. When one task is completed - new are pulled in and becomes available. The Kanban method recognizes three agendas (Anderson & Carmichael, 2016):

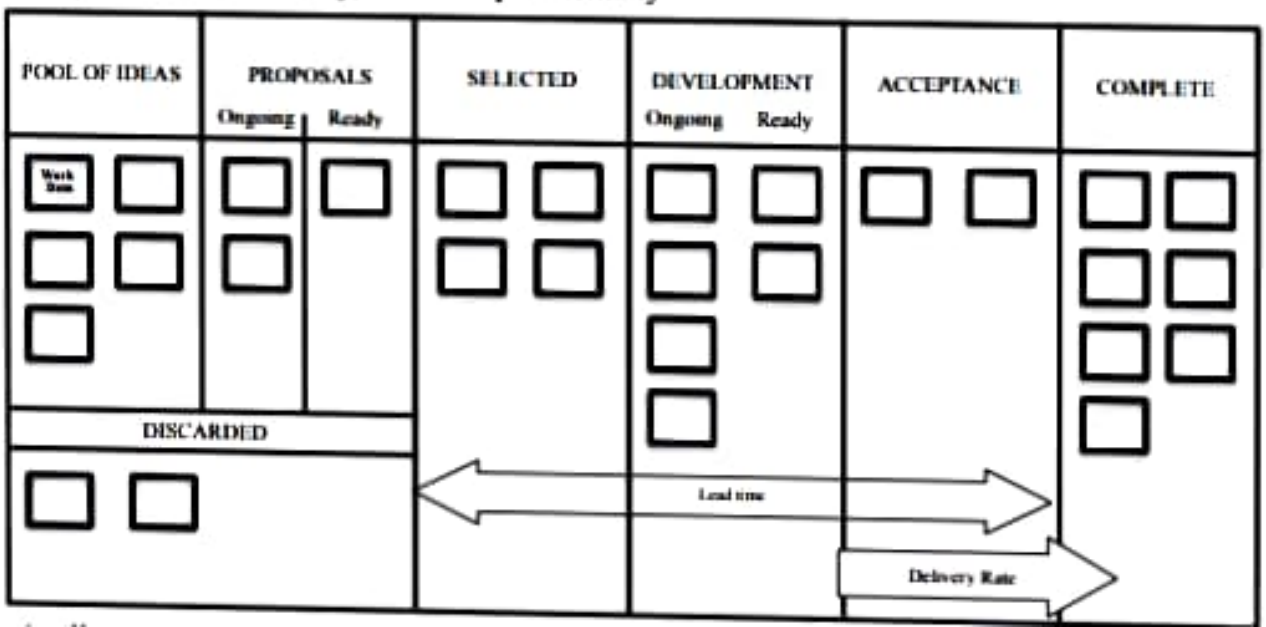
- **The sustainable agenda**
  It is about determining a good pace of work and to improve focus. Kanban method uses this agenda to ensure not too much work is put on the workers at the same time. Big focus on the balance between demand and capability with both employees and customers in mind.

- **The service agenda**
  This almost declares itself. The service agenda's big focus is on providing the customers with the best possible service and giving them great satisfaction. This is seen as a key driver to success within Kanban, and many other methods as well.

- **The survivability agenda**
  This part exist for looking out for the future and to ensure that an organization survives in the future, even when change is happening. It is all about staying competitive and adaptive to new changes and new markets.

The Kanban method consist of six parts (Anderson & Carmichael, 2016):

1. Visualize
2. Limit work in progress
3. Manage flow
4. Make policies explicit
5. Implement feedback loops
6. Improve collaboratively, evolve experimentally



AI solutions are already adopted primarily within logistics, retail management, government surveillance and manufacturing assets management. AI is forecasted to penetrate more business processes in the future and it is estimated to become an increasingly important key feature in all future business models. In short; AI is expected to be present in everything digital (Ghosh, 2018).

However, the development of any variety of information systems is not an easy task; particularly not AI solutions. In order for organizations to develop any kind of system, a standard set of steps, called a systems development methodology, is used for support and development. A widely established methodology used to describe this process is the systems development life cycle (SDLC). It features several phases that mark the progress of the system- and analysis design effort. It is common that the amount of identifiable phases varies in the SDLC based on the authors perception of the methodology, but the general content, direction and ultimate goal is the same. However, despite the variation in phases, a widely recognized-, always included- and critical phase is the requirements analysis (RA) phase. In any systems development project, the RA is described as the foundation on which the success or failure of the project relies on (Hoffer, George & Valachich, 2014).

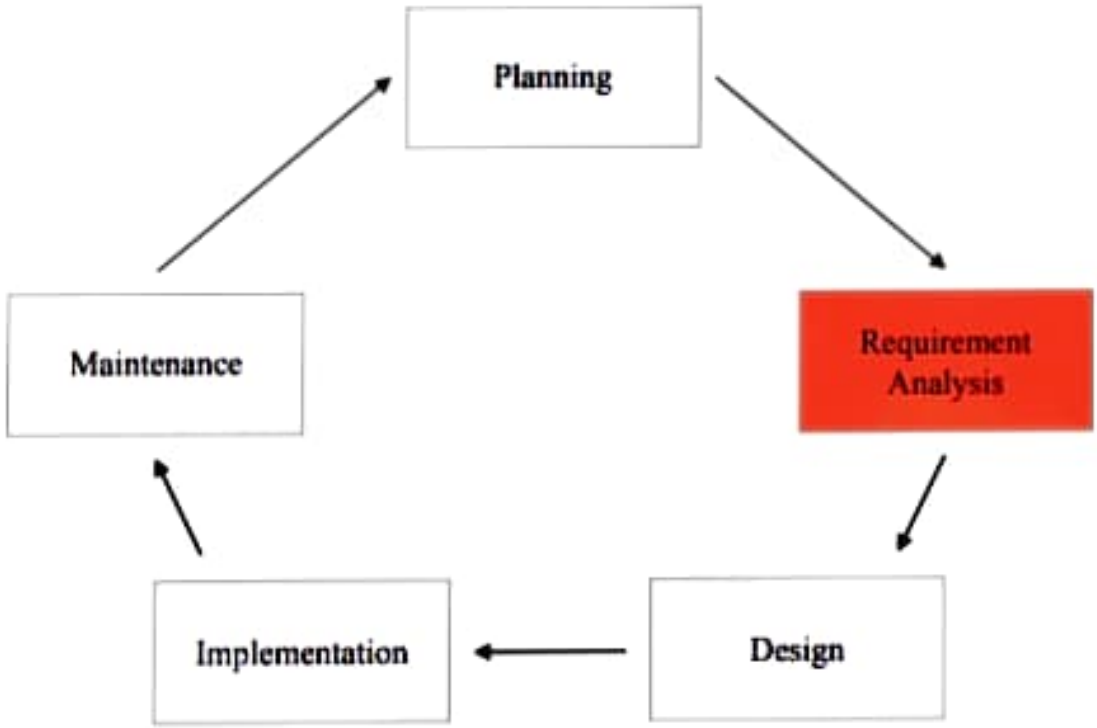


Fig. 2. An illustration visualizing the SDLC as described by Hoffer, George & Valachich (2014).

The SDLC visualized above is used to illustrate where the requirements analysis process is located in the SDLC. The goal of the RA is to determine what functions (i.e, requirements) the new system should inhabit. A requirement can be defined as a desirable function within the system. It might for example be functions that arose from an organizations desire to deal with problems in current procedures more efficiently or a desire for a system to perform additional tasks. Hence, during the RA, analysts work close together with the users to determine what the users want and need from the new system. The final step of the RA is to analyze interrelationships between the requirements in order to avoid redundancies (Hoffer, George & Valachich, 2014).

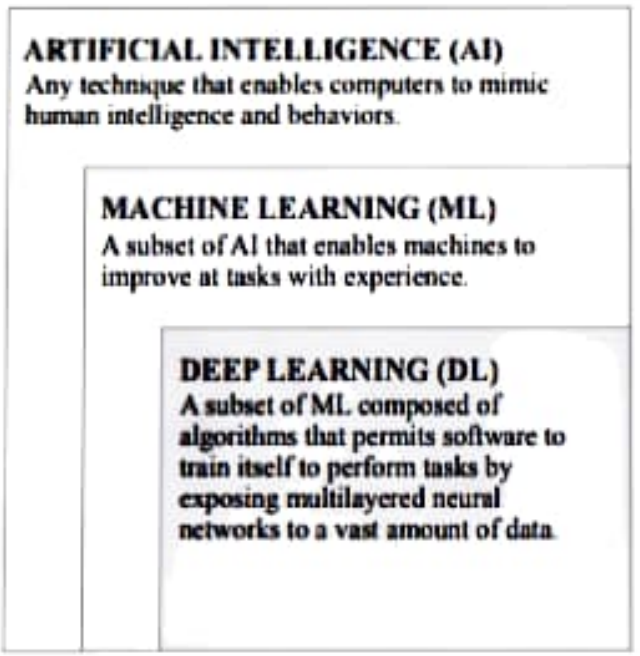
1. Introduction

In this first chapter, the research area is presented. This is done via initially introducing an overlaying background to the research problem, followed by a problem discussion, which finally results in the purpose of the study as well as the research question. Lastly, a target audience is specified.

1.1 Background

As a result of the increasingly ubiquitous world we reside in, billions of devices contribute with endless streams of information every day. As a consequence, today's globalized business environment of is vastly influenced by the importance of implementing efficient and competitive technological solutions. Typically, these technological solutions have had the purpose of managing the streams of information in order to control and understand them, for example by turning raw streams of data into useful information that can be used for decision making within the organizations. This is achieved via, for example, implementing Information Systems (IS) within organizations, which provides business analysts with assayable information (Hoffer, George & Valachich, 2014).

However, the rapid pace of technological advancement has introduced a new kind of technological solution to the market; solutions that utilizes Artificial Intelligence (AI). AI is a vast and rather obscure subject that many individuals have a hard time to grasp, yet its importance for future business competitiveness is essential, and a growing number of businesses are faced with the challenge of developing and implementing AI solutions (Ghosh, 2018). AI solutions (or AI-enabled applications and systems) can briefly be described as machines (i.e, systems) that learn from experience, adjusts to new inputs, is able to make predictions and perform human-like tasks. This is achieved via Machine Learning (ML), which is a subset of AI that can be described as an automation of the analytical process, i.e, that the system can learn from the data, identify patterns and make decisions with minimal or no human intervention. In order for ML to perform such tasks, it uses Deep Learning (DL), which is a subset or type of ML that focuses on training the computer to learn on its own (Russell & Norvig, 2010).



A poorly performed RA is often the greatest contributor to a failed project. If requirements are not well documented and set up in an efficient way, the final system will not [illegible] qualities needed from its users. Some of the consequences might be, for example, that the system development will be delayed, which in turn will make the whole project exceed the budget drastically. Furthermore, because of misjudged functional qualities, the system may risk to not perform and deliver in a satisfactory way for its users, which will lead to dissatisfaction (Eriksson, 2008).

1.2 Research Problem

Artificial Intelligence as a subject can be relatively difficult to grasp. With extensive increase in usage, interest and development of AI, numerous questions and potential dilemmas appears (Stone et al., 2016). One element in the subject of AI which is not well researched is the process of developing and implementing it as a solution into the corporate world. As with any systems development project, these AI solutions must follow a set of standardized systems development phases, such as the SDLC. This rises questions regarding how the development and implementations of AI solutions should be executed.

According to statistics in the CHAOS report from 2015, 29% of all software development projects were completed successfully, 19% were cancelled completely while 52% of projects were late and over budget. When determining the major contributors to successful projects, investments on user involvement (i.e, the part where requirements are determined by the users) was ranked in a shared first place (Hastie & Wojewoda, 2015). The statistics match theoretical assumptions on the importance of RA. For example, Eriksson (2008) describes that the cost of fixing a certain problem rises tenfold for each step in the SDLC process. A problem that costs 1000 USD to fix in the requirements phase will cost 10 000 USD in the design phase and 100 000 USD in the implementation phase, etc. Hence, It is widely recognized that the requirements analysis is a difficult yet acutely important phase for the success of any systems development project (Chakraborty, Kanti Baowaly, Arefin & Newaz Bahar, 2012).

When conducting requirements analysis it is usually done from the perspective of the users based on what functions they need and want. However, a reoccurring problem is that users often do not know what they want from the system, or if what they want is possible to implement in practice (Eriksson, 2008). Since it is already difficult for users to determine requirements for traditional systems, it arguably becomes even more challenging as they are faced with the implementation of obscure AI solutions. Not even multinational and widely established corporations with a lot of resources posses the capacity to accurately predict how AI solutions will behave.

A good example of this is Amazon. They attempted to develop an AI solution that could work as a recruitment tool to assist the company in finding the best suited people for certain job positions. Their AI solution worked by grading each applicants resume and giving them a score based on several aspects. The problem the developers found, was that the AI solution did not grade the applicants in a gender-neutral way, and rated male applicants higher than female ones. The AI solution based its grading on previous history regarding qualities by former Amazon employees from the past 10 years. Since most of the former employees were male, the system began teaching itself that male co-workers were more qualified than female

A poorly performed RA is often the greatest contributor to a failed project. If requirements are not well documented and set up in an efficient way, the final system will not inhabit the qualities needed from its users. Some of the consequences might be, for example, that the system development will be delayed, which in turn will make the whole project exceed the budget drastically. Furthermore, because of misjudged functional qualities, the system may risk to not perform and deliver in a satisfactory way for its users, which will lead to dissatisfaction (Eriksson, 2008).

1.2 Research Problem

Artificial Intelligence as a subject can be relatively difficult to grasp. With extensive increase in usage, interest and development of AI, numerous questions and potential dilemmas appears (Stone et al., 2016). One element in the subject of AI which is not well researched is the process of developing and implementing it as a solution into the corporate world. As with any systems development project, these AI solutions must follow a set of standardized systems development phases, such as the SDLC. This rises questions regarding how the development and implementations of AI solutions should be executed.

According to statistics in the CHAOS report from 2015, 29% of all software development projects were completed successfully, 19% were cancelled completely while 52% of projects were late and over budget. When determining the major contributors to successful projects, investments on user involvement (i.e, the part where requirements are determined by the users) was ranked in a shared first place (Hastie & Wojewoda, 2015). The statistics match theoretical assumptions on the importance of RA. For example, Eriksson (2008) describes that the cost of fixing a certain problem rises tenfold for each step in the SDLC process. A problem that costs 1000 USD to fix in the requirements phase will cost 10 000 USD in the design phase and 100 000 USD in the implementation phase, etc. Hence, It is widely recognized that the requirements analysis is a difficult yet acutely important phase for the success of any systems development project (Chakraborty, Kanti Baowaly, Arefin & Newaz Bahar, 2012).

When conducting requirements analysis it is usually done from the perspective of the users based on what functions they need and want. However, a reoccurring problem is that users often do not know what they want from the system, or if what they want is possible to implement in practice (Eriksson, 2008). Since it is already difficult for users to determine requirements for traditional systems, it arguably becomes even more challenging as they are faced with the implementation of obscure AI solutions. Not even multinational and widely established corporations with a lot of resources posses the capacity to accurately predict how AI solutions will behave.

A good example of this is Amazon. They attempted to develop an AI solution that could work as a recruitment tool to assist the company in finding the best suited people for certain job positions. Their AI solution worked by grading each applicants resume and giving them a score based on several aspects. The problem the developers found, was that the AI solution did not grade the applicants in a gender-neutral way, and rated male applicants higher than female ones. The AI solution based its grading on previous history regarding qualities by former Amazon employees from the past 10 years. Since most of the former employees were male, the system began teaching itself that male co-workers were more qualified than female

ones. Although the AI was programmed without any bias in mind, it began to penalize resumes where words connected to the female gender was included (Dastin 2018).

All of the aspects lifted in throughout this chapter boils down to our research problem. Throughout this section it has been established that it is already difficult to determine requirements for traditional systems, so a question naturally arises on how requirements analysis is executed as AI solutions (that even fewer individuals can grasp) are being developed. Little research has been made on how the vital requirements phase is executed during development of AI solutions.

1.3 Purpose

The purpose of this work is to explore and describe how companies manage the requirement analysis phase during development and implementation of AI solutions. The relevance and need for this work is motivated by the lack of sufficient available research previously performed within this area.

1.4 Research Aim and Questions

As the interest of AI is increasing rapidly, more and more companies are leaning towards implementing AI solutions as a tool to increase their competitive advantage and promote their business (Stone et al., 2016). Implementing AI solutions must follow a set of systems development methodologies, such as the SDLC, where one of the key phases is the requirements analysis. This particular phase is vital to ensure the success of any system development project. Therefore, our aim is to research how this crucial phase is managed when developing AI solutions.

This leads to our research question:

- How is the requirements analysis phase executed during the development of AI solutions?

1.5 Target Audience

The target audience for this study are researchers and organizations who are going to perform / performing work within the fields of this study, i.e related to requirements analysis and AI. Organizations can use this study to broaden their understanding of the subject, while researchers can use the results from this study as a springboard for future research within the field.

Before the entire project starts, the requirements for each step should be established. When leaving one phase for the next, it is difficult and can be very expensive to go back and make changes. The requirement phase will be frozen and cannot be changed. The steps above should never overlap - which means that the development should only focus on one phase at the time. One particular aspect to take into perspective in the waterfall model according to Balaji and Sundararajan Murugaiyan (2012) is that the the team who handles each phase, handles only that one and have no or very few contact with other teams. This may lead to costly problems in the end and that the system may be of bad structure when new requirements are discovered at a late stage. Some thoughts that arise here is why developers use this model if it is unchangeable and can lead to high costs when executed wrongly or poorly. There are several reasons: Requirements are set at an early stage which is easy to follow since no changes are made, it is very easy to implement since it is developed by a linear model, the amount of resources needed are very low and it consists of a high amount of documentation (Balaji & Sundararajan Murugaiyan (2012).

According to Ericsson (2008) the strive after perfection in each step of the waterfall model - may cause a lot of trouble because there are no real "perfect requirements" in real life. The longer time it takes to write and establish a perfect set of requirements, the higher the risk is that more requirements that are needed will be discovered and that the customer will therefore not be satisfied (Eriksson, 2008).

2.7.2 Iterative Development

In the sequential developing methods the system engineers tries to build an entire system all at once. According to Eriksson (2008) it is like trying to eat a whole elephant in one single meal. New kind of development methods has perspectives that differs from the traditional sequential development method. In more modern methods it is viewed as a better approach to dismember the elephant into small pieces and eat one piece at a time. This method is called iterative development (Eriksson, 2008).

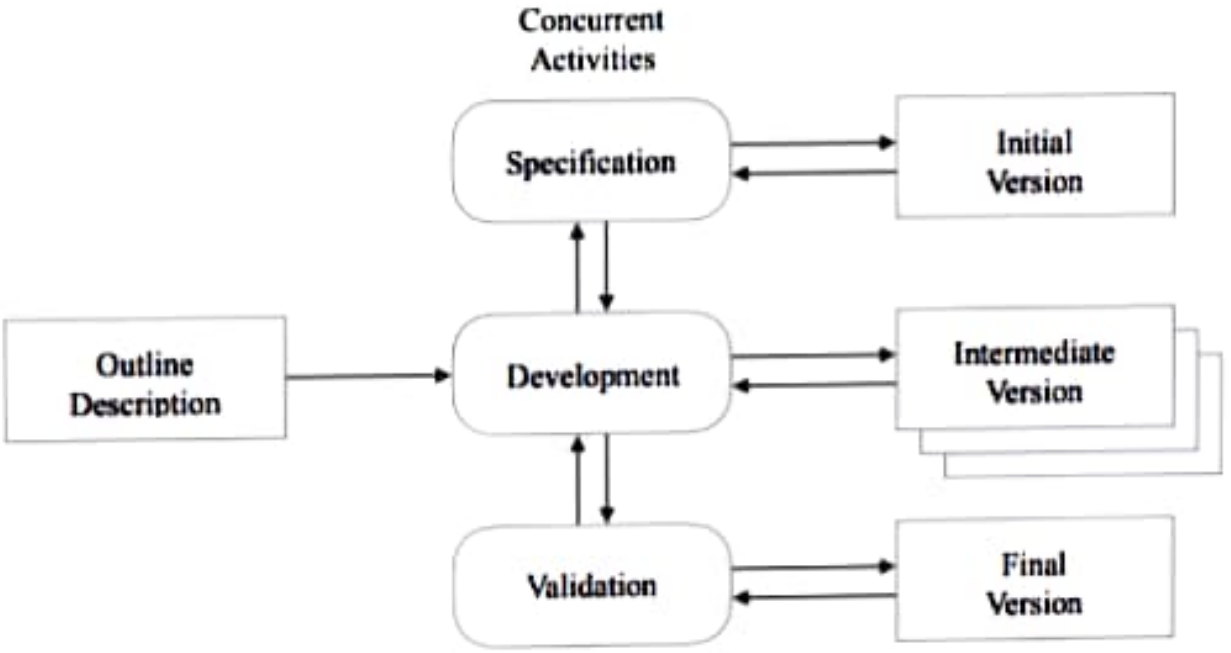


Fig. 5. An illustration visualizing the incremental model as described by Sommerville (2011).

Before the entire project starts, the requirements for each step should be established. When leaving one phase for the next, it is difficult and can be very expensive to go back and make changes. The requirement phase will be frozen and cannot be changed. The steps above should never overlap - which means that the development should only focus on one phase at the time. One particular aspect to take into perspective in the waterfall model according to Balaji and Sundararajan Murugaiyan (2012) is that the the team who handles each phase, handles only that one and have no or very few contact with other teams. This may lead to costly problems in the end and that the system may be of bad structure when new requirements are discovered at a late stage. Some thoughts that arise here is why developers use this model if it is unchangeable and can lead to high costs when executed wrongly or poorly. There are several reasons: Requirements are set at an early stage which is easy to follow since no changes are made, it is very easy to implement since it is developed by a linear model, the amount of resources needed are very low and it consists of a high amount of documentation (Balaji & Sundararajan Murugaiyan (2012).

According to Ericsson (2008) the strive after perfection in each step of the waterfall model - may cause a lot of trouble because there are no real "perfect requirements" in real life. The longer time it takes to write and establish a perfect set of requirements, the higher the risk is that more requirements that are needed will be discovered and that the customer will therefore not be satisfied (Eriksson, 2008).

2.7.2 Iterative Development

In the sequential developing methods the system engineers tries to build an entire system all at once. According to Eriksson (2008) it is like trying to eat a whole elephant in one single meal. New kind of development methods has perspectives that differs from the traditional sequential development method. In more modern methods it is viewed as a better approach to dismember the elephant into small pieces and eat one piece at a time. This method is called iterative development (Eriksson, 2008).

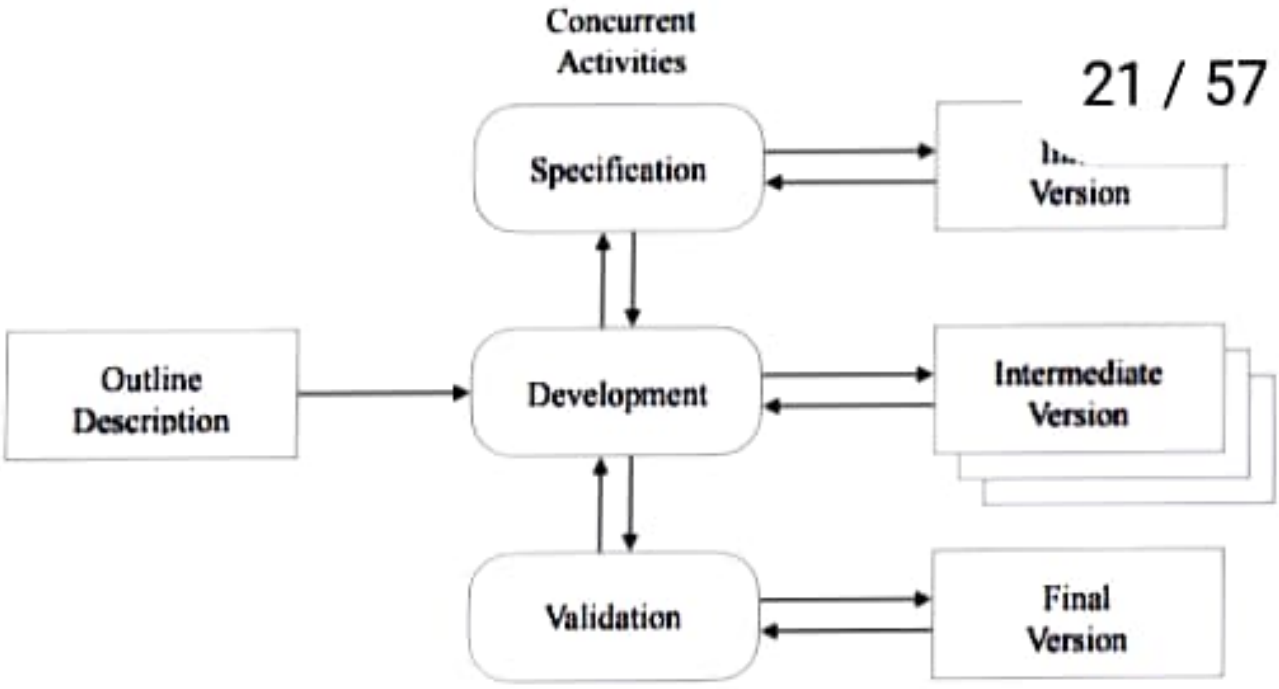


Fig. 5. An illustration visualizing the incremental model as described by Sommerville (2011).

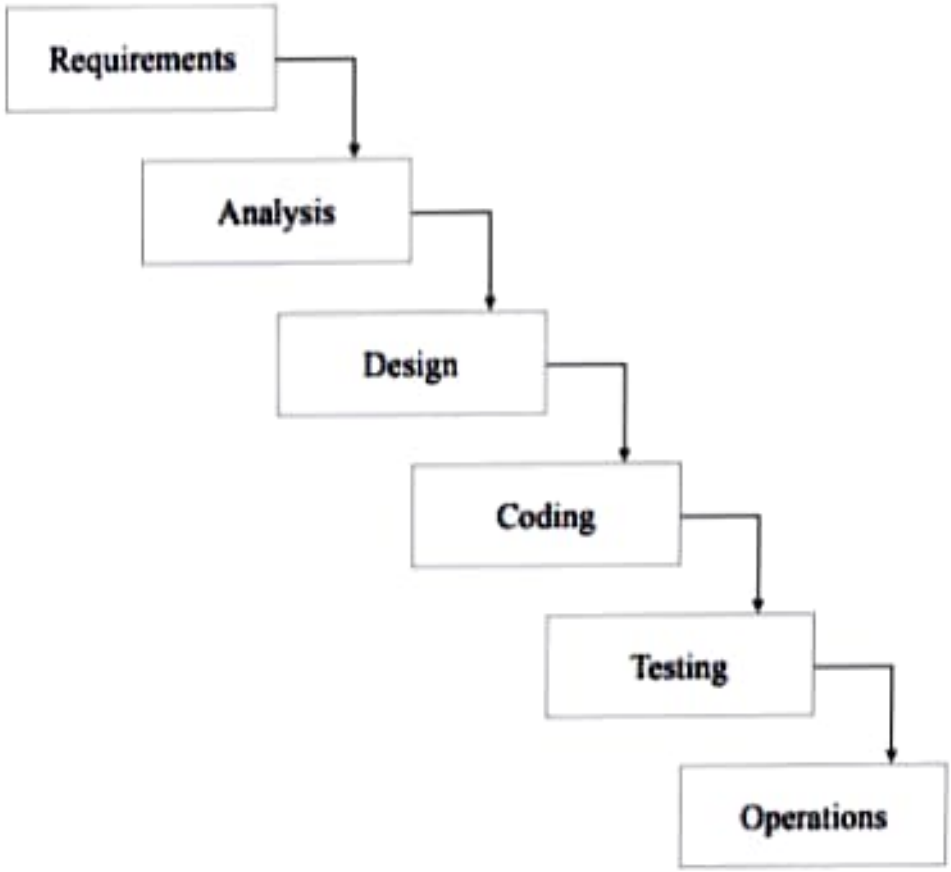


Fig. 4. An illustration visualizing the Waterfall Model as described by Royce (1970).

One of these modern adaption uses almost the same steps, but with different names and combinations. These steps are explained by Ian Sommerville (2011):

1. **Requirements analysis and definition:** This steps fundamental purpose is for the developer to establish goals, services and constraints of the system in development. All requirements are defined in detail and work as a system specification.

2. **System and software design:** This is the phase where the developer establish an overall system architecture. The design part involves identifying and describing the fundamental abstractions and their relationships in the developed system.

3. **Implementation and unit testing:** Software design is realized in this part of the waterfall model. In this stop the system has a set of programs or units. The unit testing refers to testing if all the previous specifications has been met.

4. **Integration and system testing:** All developed applications and programs are put together to form the final system. It is also tested to see that everything works together and that the earlier established software requirements are met. In the end of this step - the final system is delivered to the customer.

5. **Operation and maintenance:** Here the system is installed and maintained during the whole life cycle of the system. It involves fixing problems and errors which were not discovered earlier.

benefit of not including more organizations is that each respective organization can be profoundly examined, which will contribute to rich, deep and nuanced information that is vital for making relevant conclusions. However, since the amount of five participating organizations is relatively small, the conclusions in this work can not be viewed as fully universally representative (Bryman & Bell, 2013).

3.2 Approach

There are three major reasonings that can be applied when relating theory and empirics to each other. These are *deductive*, *inductive* and *abductive* reasonings. According to Jacobsen (2017), the *deductive* (also; top-down) premiss is that the empirical collection is based on available theory and conventional principles. In other words, the available theory decides which type of information that should be collected, how this information should be interpreted and how the result should be related to the theory (Patel & Davidsson, 2003). I.e, the process moves from theory to empiricism. This requires the researcher to have an established theory before collecting data. In contrast, when the *inductive* (also; bottom-up) reasoning is used, the theory is formed as a result of observing reality (empirics). In other words, the empirics form reality based theories, i.e the process moves from empiricism to theory. Therefore, it is of great importance that the researcher keeps an open mind when collecting data with an inductive reasoning (Jacobsen, 2017).

However, Jacobsen (2017) argues that from a pragmatic perspective, it is impossible to remain purely deductive or inductive. Instead, research is often a certain mix that contains both reasonings. Such perspective is called *abductive* reasoning. Its premiss is that all research begins by an observation of a surprising phenomena, which generates a question. The question is in turn regarded as a problem that has to be solved, which generates speculations as to why things are as they are. The speculations are then generated into hypotheses and assumptions, which naturally leads to an empirical investigation to see if the hypotheses and assumptions are correct. In short, research becomes a continuous problem solving process, which makes it a combination of both deductive and inductive reasonings. I.e, abductive reasoning is a constant interaction between theory and empirics, where no one has a higher priority than the other (Jacobsen, 2017).

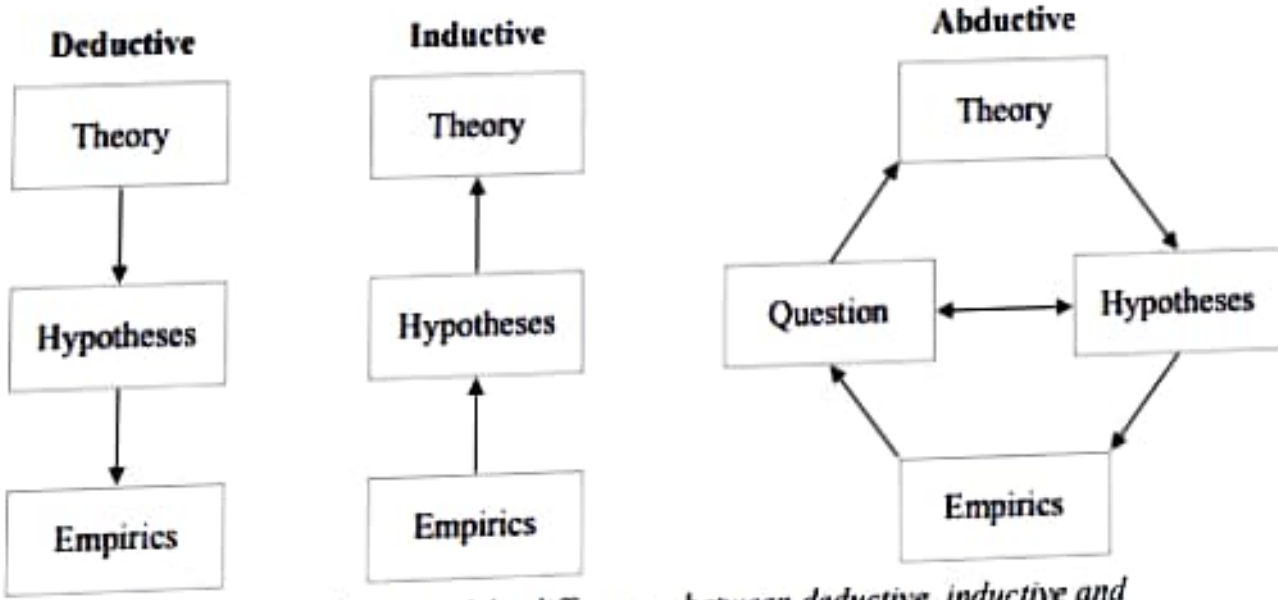


Fig. 4. A visualization of the differences between deductive, inductive and

4.2.3.2 Requirement Management Methods

Despite that requirements analysis is an inclusive phase during development, no organization used a purely specific requirements management method. Instead, requirements were typically expressed in the iterations via different frameworks, such as Kanban and Scrum. Every participant declared that an iterative and agile approach to the requirement analysis is absolutely necessary when developing an AI solution. In other words, requirements are expressed over the course of iterations, and are prioritized based on their relevance. This approach to requirements is prevailing among all the objects of study and no-one expressed that a specific method is used. Centiro, Evry and Findwise states that the requirements analysis phase differ from team to team, and that it can change over time within a specific team. The teams in those organizations are free to take parts from certain requirements management methods and mix them how ever they like.

It could also be understood that the knowledge of specific requirements terminology were mixed. Functional and non-functional requirements were not commonly known among the organisations, with the exception of one company that specifically expressed that they incorporated those terms in their daily operations. The same goes for sensational requirements, of which one organization stated that the term was well known but seldom used in practice.

Subject 1 at Centiro explains that their AI team seldom have direct communication about requirements with the end-users or customers. They explain that since the AI solution is often only a small piece of a much larger system, the teams that handle the traditional parts are often responsible for the requirement communication with the end-users and customers. To clarify, subject 1 believes that the requirements that are expressed by the end-users and customers are received by the traditional development teams, and related requirements to AI are later transferred on to the AI team. Swedavia discussed that they put together a sort of management team when receiving a request of a new system or a new function. This team communicates with the end-user to fully understand what they require from the new solution. This applies both to traditional projects and to AI projects. Furthermore, they added that the requirements analysis for AI solutions requires denser reconciliations, in comparison to traditional development.

Generally speaking, the companies explained that they have no specific method for prioritizing and structuring requirements. When asked about if they use well-established methods or techniques for prioritizing such as the MoSCoW technique - they all answered a clear no. The companies stated that they do naturally prioritize and structure the requirements, but without any specific method or technique. Instead requirements are usually expressed in different forms such as in Backlogs (for Scrum), Models, Roadmaps and through management softwares such as Trello and Jira. Through these kinds of visualizations, the customer or end-user can be a part of the development process and give new suggestions or requirements between iterations in the agile and iterative work environment. For example, Findwise describes how they use the software Trello for their requirements process and for keeping in touch with the customer. Subject 4 explains:

> Customers can log into Trello just like us... It is a web-based tool so often we sit in our status- or reconciliation meetings and go through the status of Trello