

Teemu Uusitalo

**CLOUD PRODUCT CAPABILITY
ASSESSMENT METHOD**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIETEIDEN LAITOS
2015

ABSTRACT

Uusitalo, Teemu Tapani

Cloud product capability assessment method

Jyväskylä: University of Jyväskylä, 2015. 91 p.

Information systems science, Master thesis

Supervisor: Mirja Pulkkinen

The objective of the study is to build new knowledge for product management how IT service management (ITSM) and enterprise architecture (EA) frameworks can be integrated to the cloud product management. The second objective is to build a new method called cloud product capability assessment to manage the cloud product development from inception to the go-to market situation. The method is targeted for product and project managers and also for architects working with the cloud products. In the business viewpoint a new definition for the cloud product is built, short definition of the cloud product management and need for business agility is given. In the IT viewpoint the most common ITSM and EA frameworks relevance to the cloud product management is shortly discussed. The new method development follows Design Science Research Method (DSRM) process for a new artefact building. During the method development a new agile product development framework and continuous development process for the cloud product development is built. Also a new EA viewpoint for the cloud product is created and ontological analysis of selected ITSM and EA methods is done in order to build a template for the cloud product capability assessment method. Based on the design of the method a simple toolkit is built to demonstrate the method usage. The method is used in one real life context and results are analysed. Based on the demonstration results the method is found to fulfil agile requirements for the method but visualisation needs improvements which is impacted by the technical limitations in the toolkit development. The research artefacts are made available as open source to support wider usage and future development of the method.

KEYWORDS: Product management, Product development, Product lifecycle, Cloud products, Enterprise Architecture (EA), IT Service Management (ITSM)

TIIVISTELMÄ

Uusitalo, Teemu Tapani
Cloud product capability assessment method
Jyväskylä: Jyväskylän yliopisto, 2015. 91 s.
Tietojärjestelmätiede, pro gradu - tutkielma
Ohjaaja: Mirja Pulkkinen

Tutkimuksen tavoitteena on lisätä osaamista tuotehallintaan, miten IT-palveluhallinta- ja kokonaisarkkitehtuurimenetelmiä voidaan integroida pilvipalveluiden tuotteistamiseen. Tutkimuksen toisena tavoitteena on rakentaa uusi pilvipalveluiden kyvykkyysien hallintamenetelmä, jonka avulla pilvipalveluiden tuotteistusta voidaan hallita ideasta markkinoille saamiseen asti. Uusi menetelmä on tarkoitettu tuote- ja projektipäälliköille sekä arkkitehteille, jotka työskentelevät pilvipalveluiden kanssa. Pilvipalvelulle luodaan liiketoiminnan näkökulmasta uusi määritelmä, annetaan lyhyt esittely pilvipalveluiden tuotehallinnasta sekä määritellään tarve liiketoiminnan ketteryydelle. IT-näkökulmassa tunnetuimpien IT-palveluhallintakäytänteiden ja kokonaisarkkitehtuurimenetelmien merkitys pilvipalveluiden tuotteistukseen arvioidaan lyhyesti. Uuden menetelmän kehityksessä noudatetaan Design Science Research Method (DSRM) -prosessia. Menetelmän kehitysvaiheessa luodaan uusi viitekehys ja jatkuvan kehityksen -prosessi pilvipalveluiden tuotteistamiseen. Lisäksi määritellään uusi kokonaisarkkitehtuurinäkökulma pilvipalvelulle ja tehdään ontologinen analyysi valituille IT-palveluhallinta ja kokonaisarkkitehtuurimenetelmille, jota käytetään uuden menetelmän sisällön pohjana. Menetelmän suunnitelman mukaisesti rakennetaan yksinkertainen työkalupakki, jolla menetelmän käyttöä voidaan havainnollistaa. Menetelmää käytetään yhdessä kontekstissa ja tulokset arvioidaan. Tulosten perusteella menetelmä täyttää sille asetetut ketteryyksivaatimukset, mutta tulosten visualisoinnissa on parannettavaa, mihin vaikuttaa valittujen toteutustyökalujen tekniset rajoitukset. Tutkimustulokset ja työkalut julkaistaan avoimella lähdekoodilla, jotta uutta menetelmää voidaan käyttää mahdollisimman laajasti ja se tukisi menetelmän jatkokehitystä.

AVAINSANAT: Tuotehallinta, Tuotekehitys, Tuotteen elinkaari, Pilvipalvelut, Kokonaisarkkitehtuuri, IT-palveluhallinta

FIGURES AND TABLES

Figures

FIGURE 1 DSRM-process (Peffer et al., 2007).....	8
FIGURE 2 Five levels of product (Kotler et al., 2014).....	14
FIGURE 3 Gartner hype cycle for SaaS products (Gartner, 2014a).....	18
FIGURE 4 Product manager internal and external boundaries (Geracie et al, 2013).....	20
FIGURE 5 Cloud product management.....	21
FIGURE 6 Business model role inside organisation (Osterwalder, 2004).....	22
FIGURE 7 CobiT alignment to enterprise architecture and business (IT Governance Institute, 2007).....	30
FIGURE 8 CobiT process model ontology (IT Governance Institute, 2007).....	31
FIGURE 9 Architecture Development Method (Open Group Standard, 2011).....	33
FIGURE 10 Zachman framework v3 (Zachman, 2011).....	35
FIGURE 11 Cloud Product Capability Index domain model.....	37
FIGURE 12 Capability development framework.....	39
FIGURE 13 Continuous development process.....	41
FIGURE 14 Continuous development process in the first execution.....	43
FIGURE 15 The development Kanban board.....	45
FIGURE 16 The continuous development process in the go-to market.....	46
FIGURE 17 Ontology and model analysing rules according to Meertens et al. (2012).....	48
FIGURE 18 Live events organisation.....	51
FIGURE 19 Assessment tool end user instructions.....	59
FIGURE 20 Live events status after initiation phase.....	61
FIGURE 21 Live events development Kanban board.....	63
FIGURE 22 Live events product development status.....	63
FIGURE 23 Live events change log.....	64
FIGURE 24 Survey respondents.....	69
FIGURE 25 Survey responses regarding different methods used.....	71
FIGURE 26 Survey responses for the different visualisation methods.....	72
FIGURE 27 Survey responses regarding collaboration and team level interactions.....	72
FIGURE 28 Survey results regarding usefulness of the workshop.....	73

Tables

TABLE 1 Product lifecycle mapped to product management lifecycle (Geracie et al, 2013).....	18
TABLE 2 Enterprise architecture frameworks ontological analysis.....	49
TABLE 3 Design workshop	52
TABLE 4 Planning workshop.....	54
TABLE 5 Study execution.....	55
TABLE 6 Continuous development process checkpoints.....	56
TABLE 7 Survey results statistical analysis, N = 10.....	70

CONTENTS

FIGURES AND TABLES.....	4
1 INTRODUCTION.....	6
1.1 Research method and process.....	7
1.2 Research objectives and limitations.....	10
1.3 Definitions	11
1.4 Previous researches	12
2 CLOUD PRODUCT BUSINESS MANAGEMENT.....	13
2.1 Cloud product definition.....	13
2.2 Cloud product management.....	19
2.3 Business modelling.....	22
2.4 Business capability definition.....	25
2.5 Business agility is needed.....	26
3 IT MANAGEMENT VIEWPOINT TO CLOUD PRODUCTS.....	29
3.1 IT Service Management frameworks.....	30
3.1.1 CobiT 4.1.....	30
3.1.2 ITIL v3.....	31
3.2 Enterprise Architecture frameworks.....	33
3.2.1 TOGAF 9.1.....	33
3.2.2 Zachman v3.....	34
4 METHOD DEVELOPMENT.....	36
4.1 Method design.....	36
4.1.1 Cloud Product Capability Index.....	36
4.1.2 Capability development framework.....	38
4.1.3 Continuous development process	41
4.1.4 Ontological analysis of EA & ITSM methods.....	46
4.1.5 Assessment template for the assessment tool.....	49
4.2 Method development.....	50
4.2.1 Group assessments.....	51
4.2.2 Building the assessment tool.....	56
4.3 Demonstration planning.....	64
5 DEMONSTRATION RESULTS.....	67
6 DISCUSSION AND CONCLUSIONS.....	74
6.1 Further research topics.....	76
REFERENCES.....	77
ATTACHMENTS.....	80

1 INTRODUCTION

Cloud products are so pervasive nowadays that one might not notice how many times each day they are used. One is using cloud products when reading news, sending email or communicating in real time with friends through numerous cloud based services like Twitter or Facebook. Consumers are using more and more cloud products to consume other content like music, movies or tv and radio series through cloud products. In the business world cloud products are widely used from productivity tools like Office365 to customer relationship management (CRM) and enterprise resource planning (ERP). So the cloud products are becoming to be a necessity in business process execution and management.

It's evident that the cloud products play significant role in macro economics by providing new products to the market as well as creating a lot of value for their customers and profit for cloud product companies. Gartner (2014a) predicts that already only software-as-a-service (SaaS) products can be transformational for enterprises by bringing more agility to the business processes, helping with capital outlays and increasing IT and business alignment. The cloud products' business potential and importance can also be seen when more and more businesses focuses on the cloud business model like Nokia is moving to build cloud based network management products and Finnish logistics company Posti gets more and more revenue from information logistics. Transformation to cloud business model is not always success which can be seen for example when Finnish security company F-Secure was forced to sell their security cloud product. (Järvinen, 2015; Saarelainen, 2015; Siltala, 2015).

The cloud products are nothing new in the IT market and the first economic crisis was already seen during the dot-com bust at the end of 90s. During that time unrealistic expectations were set for cloud products and their economic value creation opportunities. Also strategic thinking was neglected and it led many dot-com companies to go out of the business. (Porter, 2001.). This is natural evolution of new technologies and business models. Since that more emphasis is added to strategic thinking of cloud products when cloud products and business models have become the new norm in the IT business. This has also lead many IT companies to transform from product based business model to service business model.

People quite often think services are not products at all because services are not tangible items like products are. This is also common misconception in IT management field where service is separated from the product which is thought to be the software asset developed by the R&D function. From business management point of view the service product has own characteristics compared to tangible products and the service product needs more holistic view how the value is created through the company (Kotler & Armstrong, 2010). The cloud products are information technology products which needs to be managed like any other service products. There is no well-knowns frameworks publicly available to manage cloud product development which is the key driver for this thesis.

The cloud product definition is created based on business and IT viewpoints in order to have a clear definition what is a cloud product and what kind of special characteristics the cloud product has. These characteristics must be taken into account in the cloud product development. The cloud product characteristics are then reflected to product management principles in order to highlight what the cloud product needs from the product management. The need for business agility is also reviewed from the cloud product perspective.

In order to build a holistic view for the cloud product management IT service management (ITSM) and enterprise architecture (EA) frameworks are considered to have broad body of knowledge which can be utilised in the cloud product management. The objectives of the study is to build new knowledge and integrate the most common ITSM and EA frameworks to the cloud product management. The integrated viewpoint to the cloud product management is utilised to build a new method called *cloud product capability assessment* to manage the cloud product development from inception to the go-to market situation. The method is targeted especially for product managers, architects and project managers working with the cloud products. The integrated viewpoint to the cloud products works also as a general guidebook to the cloud products and the cloud product management.

The cloud product capability assessment method development follows Design Science Research Method (DSRM) process for a new artefact building. During the method development a new agile product development framework and continuous development process for the cloud product development is built. Also a new EA viewpoint for the cloud product is created and ontological analysis of selected well-known ITSM and EA methods is done in order to build a template for the cloud product capability assessment method. Based on the method design a simple toolkit is built to demonstrate the method usage.

The cloud product capability assessment method usage is demonstrated in one real life context and the results are analysed. Based on the results the method is found to fulfil agile requirements for the method but visualisation needs improvements which is impacted by the technical limitations in the toolkit development. The research artefacts are made available as open source to support wider usage and future development of the method.

1.1 Research method and process

Traditional descriptive research theories, paradigms and methods for example in natural science focus on analysing the world and phenomena from the viewpoint what the world is and how it works. Natural science research uses different methods like controlled tests, surveys and case studies to test and validate existing theories and paradigms. Typically new scientific information is created when theories can be proved to valid or invalid which then drive the future research to formulate and test new theories. (Järvinen, P., & Järvinen, A., 2011.).

Information systems (IS) research is often thought to be applied science which uses basic theories for example from natural science research and apply it to IS. Järvinen et al. (2011) and Peppers, Tuunanen, Rothenberger and Chatterjee (2007) claim that aforementioned statement of IS research undervalues the value of IS research and

argue even that it might be the reason why IS researchers' focus stays in engineering disciplines like requirements and software engineering. IS research target is typically to build new innovations and analyse what the situation could be not what it is. Therefore the research target is constantly changing and the research focus is on building new constructions to solve a specific problem. This construction building research is called design science (DS).

IS construction building and design science is done typically at the intersection of information technology (IT) and organisation. Therefore it is natural that IS research applies theory from other disciplines like economics, computer science and the social sciences. Both Järvinen et al. (2011) and Peffers et al. (2007) have analysed multiple DS research in the IS domain and promote their own DS framework and methodology. In this research Peffers et al. (2007) Design Science Research Method (DSRM) is used because it provides more general framework for construction building. Järvinen et al. focus in their framework more on implementation of a new IT system when this research objective is to build a new IS method for product management purposes.

Figure 1 shows the DSRM process. According to the DSRM process there is four different research entry points. (Peffers et al., 2007.). The researcher acknowledges the research started originally roughly in the year 2010 from the problem that business and IT people don't have a common way to talk and build the cloud products. This can easily lead to the situation where benefits are not realised from the IT investments, see e.g. Peppard, Ward and Daniel (2007) and lack of alignment in the organisation, see e.g. Henderson and Venkatraman(1999).

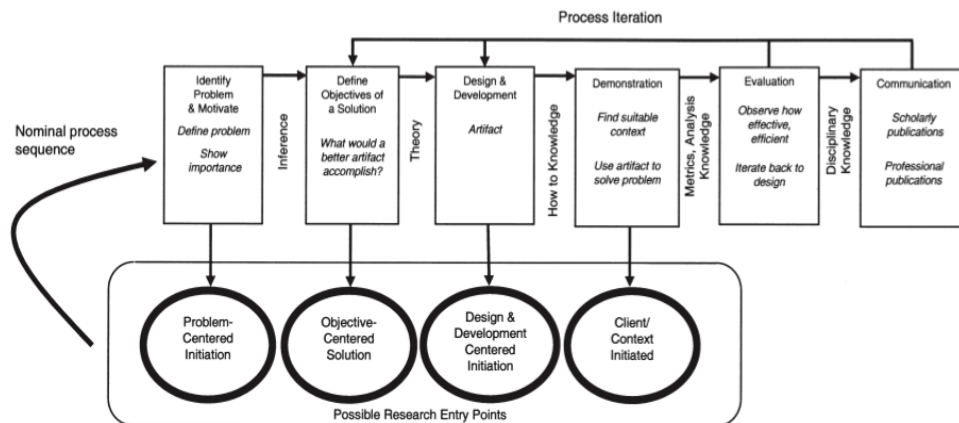


FIGURE 1 DSRM-process (Peffers et al., 2007)

Like the DSRM process suggests the research problem was relatively soon turned to objectives of a solution. The researcher iterated the problem and the objectives of the solution between 2010 and 2013. During that time the problem definition moved from business process capability assessment method to more holistic product capability assessment method. This was impacted by the change in the researcher profession as well as increased knowledge from the theory.

Research problems

Research problems are listed and categorised below. The first research problem is to understand how different business and IT theory, methods and frameworks define cloud product, capabilities and their relations. This is done as a literature review of different

theory, modelling tools, methods and frameworks which are relevant for cloud products. The literature review is split into two different sub-problems: the first sub-problem focuses on analysing business modelling, product concept and product management in relation to define capabilities for the cloud product. The second research problem in the literature review focuses on analysing how well-known Enterprise architecture and IT Service Management frameworks can be used to define the cloud product capabilities.

The second research problem is split to three sub-problems. The sub-problems are answered by building a new method called *cloud product assessment* to analyse a cloud product, needed capabilities and how capability development is managed from inception to go-to market. Finally the method is empirically demonstrated in a case study in order to validate the method feasibility.

1. What kind of capabilities a cloud product needs?
 1. What kind of capabilities business theory provides for the cloud product?
 2. What kind of capabilities EA & ITSM methods provide for the cloud product?
2. How to manage the needed capabilities through the product life cycle from inception to go-to market?
 1. How to analyse needed capabilities?
 2. How to follow progress during the product development with needed stakeholders?
 3. How to involve and get commitment from the different stakeholders to build the needed capabilities?

Research process

The research problems, objectives and limitations of the research are described in the chapter 1. Chapter two and three describe academic background for the research in the form of a literature review. In the chapter two research objectives and problems are looked from the business point of view. It is also defined in the chapter two what is the cloud product and product management to frame the context of the research. This chapter gives an overview of the cloud products for product management and IT professional. In the chapter three common IT service management (ITSM) and enterprise architecture (EA) frameworks are shortly described and their relation to the cloud product concept and management analysed. This chapter gives a basic overview of ITSM and EA frameworks and principles for product management professionals in the context of the cloud product capabilities.

The chapter four describes how the product capability assessment method is built based on the different theories and methods. Some of theories are described in more details in the literature review. Ontological analysis is utilised to map different ITSM and EA frameworks together. The way to demonstrate the cloud product capability assessment method is also elaborated in the chapter four. The chapter four maps with the DSRM process artefact design and development phase.

According to Peffers et al. (2007) the demonstration phase in the DSRM process is important in order to show how a new artefact can solve one or multiple instances of a problem. The cloud product capability assessment method demonstration results are covered in the chapter five. The chapter six focuses on the DSRM process analysis phase where the cloud product capability assessment method is evaluated from

the research objectives point of view. The analysis is based on quantitative and qualitative data elaborated in the chapter five. In the analysis phase also future research ideas and development topics are given. The final process phase in the DSRM process is communication which is covered by this research report.

1.2 Research objectives and limitations

Research objectives arise from the researcher experience on IT and business management. The first objective of the research is to build new knowledge on product management how ITSM and EA frameworks can be used to support product management, especially the cloud product management. Target is also to integrate business modelling with EA and ITSM frameworks which means building broader understanding how these different viewpoints relate and can be integrated with each others.

The second objective of the research is to build the *cloud product assessment method* to support the cloud product development. The method should be easy to use, agile and speed up cloud products productisation over the whole product life cycle. This research focuses on the product life cycle phase from inception to go-to market but the method should scale to cover the other product life cycle phases as well. The inception to go-to market phase in the product life cycle is considered be the most critical, laborious and risky part of the product life cycle and therefore it needs special attention and care.

Easy to use means the method should be simple to understand and easy to adapt to different contexts. Agile means the method should be compatible with Agile manifesto values (Manifesto for Agile Software Development, 2001). The method agile characteristics and values should be taking individuals and interactions into account, focus on working product, increase customer collaboration and understanding and respond to change. Speeding up the cloud product productisation means the method should be able to highlight what kind of capabilities are need, how to manage capability development and manage risks. These characteristics should speed up and make capability development less error prone exercise in the organisation.

The method is targeted for product managers, project managers and architects working with the cloud products. The method should work as communication mechanism for the product team, development teams and the key stakeholders what is needed to get a new product to the market and what is the new product development status. Objectives of the method are highly ambitious and the real success of the method is out of this research scope. Eventually the target audience will show if the research has been successful and they are willing to use the method in different contexts. The success criteria of the thesis is that it follows defined research process and can be demonstrated in a real life context.

One of the success criteria for the method is also if it manages to expand agile, namely Agile manifesto (2001), values and way of working outside of the software development context. This should increase cross-functional collaboration inside organisation, improve product quality and general productivity like the agile development methods have done for the software development and teams, see for example Lessons in Agile management (Anderson, 2012) or Agile Service Development (Lankhorst, 2012a).

Limitations

All academic research has limitations and this research is no exception. The researcher acknowledges the frameworks used in the research has broad body of knowledge. All the ITSM and EA frameworks are introduced in the literature review only on high level in order to highlight why they are important for the cloud product managers and the cloud product management. Researcher suggests further readings for each topics if the reader wants to build deeper understanding of the framework or method at hand.

Another limitation is the used version of CobiT framework. During the thesis writing the CobiT framework is updated to the next version but the researcher couldn't get this version for evaluation. In the demonstration phase details like the capabilities under evaluation can't be published because they contain classified information of new products. The researcher doesn't think publishing of the details of the capabilities or assessment results are relevant from the design science and research objectives point of view. The design science focuses on building new artefacts and demonstrating how those can solve common problem and if desired target is achieved (Järvinen et al, 2011). The research doesn't cover either implementation of individual capabilities because there is many different theories, frameworks and methods available from software engineering to organisational change management.

1.3 Definitions

Architecture

Architecture has the official ISO definition which defines that architecture is structured definition of system. TOGAF framework extends this concept and takes into account definitions management and whole life cycle of architecture. (Open Group Standard, 2011). In this research TOGAF's broader definition of architecture is used. Enterprise architecture (EA) is used only to refer to enterprise architecture frameworks.

Service

Service is considered to be a cloud product developed, maintained and offered by the organisation to the market. TOGAF broader definition of business service is not used because it might cause confusion compared to IT service management frameworks (Open Group Standard, 2011).

Capability

Capability has multiple different definitions and attributes like system or organisational ability (Open Group Standard, 2011), process maturity (Löhe and Legner 2013) or entity ability in business modelling (Jacob et al. 2012). In this research capability is considered to be an intangible asset of a service product which creates value for the organisation. Organisation has therefore economic interest to manage capabilities and their life cycle.

Product development and life cycle

Product development is typically considered in software industry to be R&D process where software artefact is developed using different software development methods like

waterfall or Scrum. Product life cycle is typically also mixed with systems development (SDLC) or application development life cycle in the software industry. In this research product development and life cycle is considered to be all activities needed in the organisation to build products and services to the market. SDLC is considered to be part of product development and life cycle, especially important for cloud products.

Product team

Product team is a cross-functional team who is involved and committed to do product development in the organisation.

Cloud product

There is no one single definition of what a cloud product is. In this thesis a cloud product is considered to be a product which fulfils National Institute of Standards and Technology definition of cloud computing and is offered to market to satisfy a want or need (Mell & Grance, 2009). This definition includes also Kotler et al. (2010) definition what a product is. Broader definition for a cloud product is given in the chapter 2.1.

1.4 Previous researches

Correia and Abreu (2009) have done research where IT service management is taken into enterprise architecture context. Their research approach was targeted to quality analysis, SLA compliance and they have built conceptual framework which combines IT service management and enterprise architecture to Model-Driven Development (MDD).

Nabiollahi, Alias and Sahibuddin (2010) have studied ITIL v3 and TOGAF adoption in different organisations. Their conclusion is that organisations which have implemented TOGAF in use will try to implement ITIL or other IT service management frameworks in use and vice versa. Their target was also to design an integrated framework in further studies which combines aforementioned frameworks.

Meertens, Iacob, Jonkers, Quartel, Nieuwenhuis and van Sinderen (2012) have used ontological analysis method to analyse business model canvas and TOGAF enterprise architecture framework. Research conclusion was that ontological mapping can be used to map these to different ontologies together. In the research Meertens et al. (2012) utilised TOGAF Archimate language to map business model canvas to TOGAF ontology.

Multiple other researches show initiatives to map enterprise architecture to IT service management. Sun, Xiao, Bao and Zhao (2010) have developed own architecture model to manage cloud services. Valiente, García-Barriocanal and Sicilia (2011) have developed software engineering model to integrate IT service management to software development processes. Nabiollahi, Alias and Sahibuddin (2011) suggests mapping IT service management to enterprise architecture by utilising knowledge management repositories and principles. There is also multiple meta-models developed around combining IT service management to enterprise architecture frameworks like ADRIMA method (Löhe et al., 2013) or Integration of IT Service Management into Enterprise Architecture (Braun & Winter, 2007).

2 CLOUD PRODUCT BUSINESS MANAGEMENT

This chapter is an introduction to the cloud products from business and product management point of view. A definition for the cloud product is given which will be used through the thesis. A short introduction of the cloud product management and the cloud product manager role in the organisation is also given which explains the context where the cloud product capability assessment method should be used.

Business modelling is the key area in the business management. The business model relation to the product concept is also shortly discussed. Finally analysis of the need for the business agility in the cloud business and business agility relation to agile development methods is given. Throughout the chapter basic product and business theory is reviewed from the cloud product point of view in order to distinguish the cloud product from other product types and highlight any specialities the cloud products have.

2.1 Cloud product definition

Before definition for the cloud product can be given it's important to understand what a product is. When talking about cloud products people generally tend to have misconception they are not products at all because they are services which is mentioned also by Geracie and Eppinger (2013). This misconception is most likely caused because there is no well-known and simple definition for the product.

The broadly accepted and known definition of the product is made by Kotler et al. in their famous book *Principles of Marketing* (2010). Kotler et al. introduce five levels of the product which are shown in the figure 2. The core benefit is the reasoning why the customer is looking for different products and it's the most important buying factor. Kotler et al. rationale is that product has no intrinsic value but customer is seeking benefits that satisfy some specific need or want. If the product is not delivering the core benefit the customer will most likely reject the product (Geracie et al., 2013).

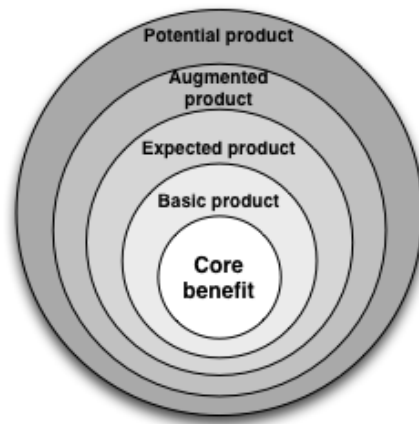


FIGURE 2 Five levels of product (Kotler et al., 2014)

The basic product is the actual product or service the customer is buying. This layer differentiates product from competitors with a defined set of attributes and features. The expected product is the customer expectation of the product and the company selling it, which can be, for example, expected speed or quality of the product. The augmented products are the product capabilities which take the product beyond the expected level, with or without extra charge, like delivery to home. The potential product describes what the product can be in the future with extensions to the product or customisations for the customer. The product capabilities move between the product layers over time because of competition and competing force changes. (Kotler et al., 2010.).

An example cloud product case is introduced in chapter 4.2 where an event organising company starts building a new cloud product to maintain and increase revenues. One aspect of the example case is an electronic ticketing system. Next, an electronic ticketing system is considered as a standalone cloud product which is compared to product definition. Before defining core benefits of the ticketing system, the target customer must be defined. The target customer is not typically a consumer but a company who is organising events and wants to sell tickets for events to consumers or other companies.

Based on that, the core benefit can be considered to be online ticket sales for the organiser. The basic product could be then an online ticketing platform where consumers can easily find and buy tickets for the events and the organiser can setup events easily by themselves. The expected product could be a reliable and trustworthy buying process with major credit cards for consumers and easy money transfer to the organiser. The augmented product could be sending printed tickets to the consumers without extra fee. The final layer of the cloud ticketing system could be a mobile application for consumers to follow their favourite artists or a way to setup multiple events by the organiser for the same artist in different countries in the future. The potential product can be something which is built later on to the cloud ticketing product as an extension.

As the examples show, the cloud product can have multiple aspects which go beyond the application itself but are necessary from the product point of view. There are multiple value aspects in the product for the customers and those are generally called

as the product value proposition. The value proposition is the perceived value the customer gets by using the product compared to perceived costs of using the product. Generally saying the product value proposition is positive if the customer thinks she gets more value out of the product than being without the product. (Geracie et al, 2013.). In the example ticketing system case it would mean it's cheaper for the organiser to use an external ticketing system including also costs of using the product than selling the tickets without the ticketing system for example through their own online ticketing system.

The online ticketing system is also a good example of a product which has tangible (tickets) and intangible (the service) parts. Intangibility is one of the characteristics of the service and means the service doesn't result in ownership of anything. Other aspects are perishability, inseparability and variability. Perishability means the service can't be stored for later use. Inseparability means the service can't be separated from the provider whether they are machines or people. Variability means the service quality vary by who provides the service as well as how and when the service is used. (Kotler et al., 2010.).

These service product specific attributes can also be seen for example in SaaS-QUAL method for measuring software as a service (SaaS) product quality. In the SaaS-QUAL method application features are only one factor which impact the perceived quality of the service. Other aspects are rapport, responsiveness, reliability, flexibility and security/privacy. For example rapport means the service provider ability to assist the customer with any needs they have and responsiveness means the service provider ability to ensure availability and performance of the product, including the availability of the support. (Benlian, Koufaris, Hess, 2010.).

Based on this it is clear that cloud products can have many other aspects than just the technical cloud service. This results to the conclusion the cloud product should be considered as mixture of goods and services which is typical for all products (Geracie et al, 2013). Other aspects can include for example professional services to assist taking the cloud product in use or warranties like service availability with agreed service levels, typically included in the service level agreement (SLA). In order to call a product to cloud product the vast majority of the core benefits should be delivered via the cloud service. For example a company assisting other companies to take cloud products in use shouldn't be called as a cloud company but a professional services company and their product is the professional services they can provide.

It's worth noticing that companies, especially enterprises, rarely have only one product on the market but multiple products or product lines. Product line or portfolio management is not in the scope of this thesis. Generally portfolio management is all activities in the organisation needed to optimise all products and their life cycle through segmentation and product development prioritisation (Geracie et al, 2013).

Cloud product definition

In the definitions section a cloud product is defined to be a product that fulfils National Institute of Standards and Technology (NIST) definition of cloud computing (Mell et al., 2009). Next NIST definitions are elaborated and compared to the product definition above in order to enclose the cloud product definition.

On-demand self-service is according to NIST (Mell et al., 2009) provisioning of computing capabilities automatically without requiring human interaction. Hill, Hirsch, Lake and Moshiri (2013) enrich the definition by saying this should support

agility and autonomy as well because resource provisioning have had significant lead time in the past. In the context of products the provisioning might have pricing or contractual implications. Therefore cloud product definition should be ***on-demand and mostly self-service*** which means there can be manual interventions like contract negotiations or provisioning might need further assistance from the vendor e.g. in form of configuring the solution for the customer according the customer needs. The latter case is common in case of complex cloud applications where customer might not have interest to learn how to do configurations. Service provider can also limit customer possibilities to make changes in order to sell more of own or partner network professional services which is part of product business model. Implicit to the definition is that underlying computing resources can be scaled on demand basis and they don't need manual intervention.

Broad network access is the next essential characteristic in NIST classification which means resources must be accessible over network such as the Internet and resources can be accessed through standard mechanism. Standard mechanisms means that resources can be accessed with e.g. tablets, mobile phones or personal computers (Hill et al, 2013). From cloud product perspective this definition doesn't take into account other resources like humans or tangible products which can be part of the cloud product concept. The cloud product definition should therefore be ***broad network access of core assets*** which provides the flexibility the product to have other assets like human resources, tangible products or even other services included. Implicit to this definition is that core benefits are delivered via network access and other aspects of product makes access possible or are otherwise needed to fulfil the product concept.

Resource pooling is defined in the NIST classification to mean that resources are shared to serve multiple customers and individual customer doesn't generally have control or knowledge of resources location, physical or logical. Basically from customer point of view this means resources seem to be homogeneous and can scale infinitely based on the customer demand. This is not necessarily the case when the cloud product includes for example so called ring fenced resources like project manager or named individual experts like consultants for managed services, see e.g. Guide to cloud computing (Hill et al., 2009). The NIST definition can easily be expanded to cloud products by defining **general resource pooling** which now includes specific aspects of the cloud products from business perspective.

Rapid elasticity is according to the NIST almost unlimited ability to scale out or scale in the capabilities needed. When taking other definitions of the cloud products like professional services resource pooling into account this is hard to achieve. Therefore the definition should be expanded to **rapid elasticity of core assets** which limits the requirement to only to the cloud service aspects like with the broad network access requirement.

The final definition in the NIST classification is **measured service** which demands that the service has capabilities to report, control and monitor the service usage with appropriate meters. Measuring is needed in order to provide transparency for both the service provider and the user. This is fully applicable for the broader definition of the cloud product and measuring needs to take into account all resources including the human resources the customer has ordered.

These characteristics are applicable to all different *service models*: Software-as-a-service (SaaS), Platform-as-a-service (PaaS) or Infrastructure-as-a-service

(IaaS). In SaaS model consumer mostly only uses the application via thin client like web browser. In PaaS model consumer utilises supported tools and programming languages to deploy their own applications. IaaS model is the closest to actual technical infrastructure where consumer manages computing resources like storage and computing power to deploy and build own services. Implicit to all service models is that the consumer doesn't control or manage the underlying infrastructure but consumes needed resources. (Mell et al., 2009.).

NIST service model classifications and definitions are also applicable for broader definition of the cloud product because core assets can be mapped to one or many of the service model categories. It's essential to understand which is the selected service model for a cloud product because it impacts heavily on business model and needed capabilities. One cloud product can though support multiple service models for example being a SaaS solution for one customer segment and a PaaS solution for another customer segment. This can also be seen in Gartner's (2014a) Hype cycle for Software as a service report where multiple PaaS solutions are shown in adjacent with SaaS solutions, see figure 3 for details. The report also reveals that multiple SaaS vendors are providing more and more PaaS like capabilities for example APIs (Application Programmable Interface) to implement governance processes, in order to support enterprise needs.

Deployment models are also one of the key characteristic of the cloud product. There is four common deployment models available: *Public*, *Private*, *Community* and *Hybrid*. Public cloud means that the service is available for all consumers and share all resources. Private cloud means that resources are assigned and allocated to a specific customer and they are not shared among other customers. Community cloud is a private cloud which is shared between common interest group like government entities. Hybrid cloud model is a composition of at least two different cloud deployment models which are integrated together. (Hill et al., 2013.).

Service and deployment models are essential characteristics of the cloud product and it's important to recognise the differences between these models. All models can also impact different layers of the product for example APIs in a SaaS solution can be considered to be the augmented product. It's important for the cloud product manager to understand the differences between all the models and when different models should be implemented. All of the models impact to the product business model and available business opportunities but they also have significantly different requirements for the capabilities needed and the requirements needs to be addressed though the product life cycle. All of the models can't be described in details in this thesis but reader is referenced to Guide to Cloud Computing (Hill et al., 2013) which explains the models and differences between the service and deployment models in more details.

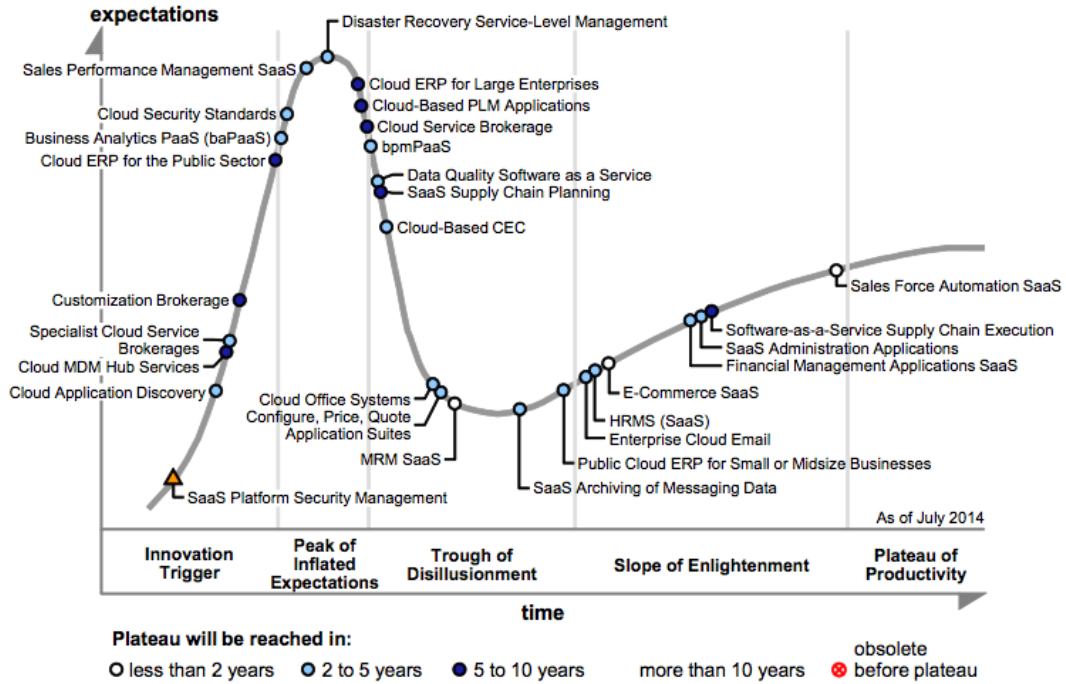


FIGURE 3 Gartner hype cycle for SaaS products (Gartner, 2014a)

Product life cycle

The product life cycle is a general concept explaining generated revenue and customer expectations for the product over time and it has four main stages: *Introduction*, *Growth*, *Maturity* and *Decline* (Kotler et al, 2010). Different phases also target to different kind of customers from innovators and early adopters to laggards and each customer segments has different expectations for the product. Sales also increases over different phases until it starts finally decreasing in the Decline phase. Majority of the products are in the *Maturity* phase where the vast majority of revenue is generated through incremental innovation with better market segmentation. (Geracie et al, 2013.)

Product management lifecycle is according to Geracie et al. (2013) process to manage the product through various phases in the product lifecycle. The product management lifecycle has seven phases: *conceive*, *plan*, *develop*, *qualify*, *launch*, *deliver* and *retire*. The product management lifecycle maps to the product lifecycle and the mapping is shown in the table 1.

TABLE 1 Product lifecycle mapped to product management lifecycle (Geracie et al, 2013)

Product lifecycle	New product development				Introduction	Growth/ Maturity/ Decline	Withdrawal
	Conceive	Plan	Develop	Qualify			
Product management lifecycle	Conceive	Plan	Develop	Qualify	Launch	Deliver	Retire

In this thesis product life cycle is considered to follow Kotler et al. (2010) definition of the product life cycle and the focus is on a new product development phase. This phase has two major milestones which are *Inception* and *Go-to market*. Inception is the starting point of a new product development where the idea of a product or business opportunity is recognised and started to be refined as the product concept and the product development starts. This maps to the conceive phase in the product management lifecycle. When the product development is done the product is considered to be in the go-to market milestone and ready for market introduction. The product development overlaps all phases from conceive to qualify in the product management lifecycle. The product development includes all the capabilities needed for the product in go-to market situation. In the later phases of the product life cycle more capabilities are most likely added to fulfil all product aspects.

The reason why the product development in this thesis is considered to overlap all four phases in the product management lifecycle is the need for agility in the cloud product development. Geracie et al. (2013) describe that the product management lifecycle is serial process with major gates but they also recognise other product development approaches, namely Minimum Viable Product concept and Agile product development. For additional information why the cloud product development needs agility please see chapter 2.5. In order to understand all the expected deliverables and tasks in the product management lifecycle the reader is referenced to The guide to the Product management and marketing Body of Knowledge (Geracie et al, 2013).

2.2 Cloud product management

Product management is the organisational function and process focusing on managing the products in the organisation, typically in a larger enterprise. The main responsibility of the product management is to ensure products and product portfolio profitability through the product life cycle as explained in the previous chapter. The product management is meant to balance internal and external demand and focus on the most profitable development projects in order to ensure market success. (Geracie et al, 2013.).

The product management has two focus areas: internal and external which can be seen in the figure 4. Internal focus area means internal teams, processes, functions and management. External means markets, supply chain, distribution channels and other relevant parties like market research companies and legislators. Internal and external focus areas are heavily dependent on the product management structure, company industry and which stakeholders mostly impact on defining, developing and delivering the product. (Geracie et al., 2013.).

According to Geracie et al. (2013) the product management performs best when it has the autonomy to balance internal and external demands. Typically there is rivalry between short term needs for example in sales and support organisations compared to long term needs and benefits for the customers, market and organisation. The product management is tasked to balance these needs and drive market success.

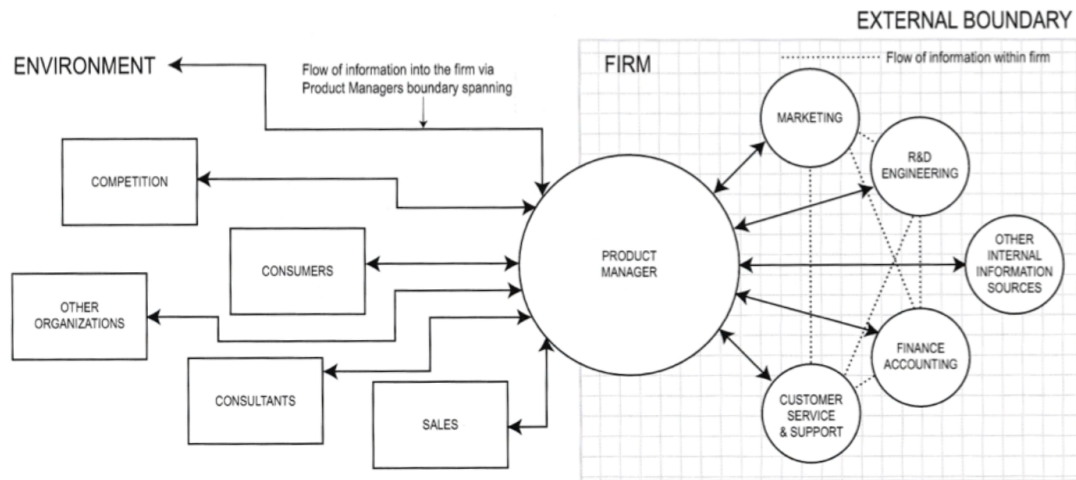


FIGURE 4 Product manager internal and external boundaries (Geracie et al, 2013)

One common misconception is that the product management is general management of the whole company. This is not typically true even though product management can have for example overall responsibility of profitability, sales channels, delivery, operational roll-outs and support of the product which means the product management being general management of the product through it's life cycle. The product management is also the only function being involved through the whole product life cycle from the inception to the retirement of the product. The major difference to the general management is that the product management doesn't have full profitability responsibility and neither functional leaders report to the product management. The product management interact with the general management by influencing and relying on the shared objectives. (Geracie et al, 2013.).

Based on the influential role of the product management it's important to understand the product team behind the success of the product. The product management doesn't typically have authority to make all the needed decisions in the organisation. Based on that it's important for the product manager to define who should participate on the cross-functional product team to drive the new product development as well as maintaining the rest of the product life cycle. There is also many stakeholders interested in participating to the product development but the product manager needs to find how to best balance customer needs to the organisational capabilities. (Geracie et al, 2013.). Therefore it's hard to define exact guidelines for the right composition or organisational functions needed in the product team for the cloud product capability assessment method explained later in this thesis. It's assumed in the method development that product manager knows the organisation and the key stakeholders needed for the product development in case there is no formal management system in the organisation which would enforce the product team structure.

The product manager works in cross-functional setting and therefore it's important that the product manager has broad understanding of different competence areas in the industry she or he is working on. For the cloud product management and generally software product management the most important knowledge and competence

domains are shown in the figure 5. The product manager works in the intersection of business, technology and design domains but the focus factor can vary between the different product manager role inside the organisation. In order to understand all the different product manager roles in the product management the reader is referenced to The guide to the Product management and marketing Body of Knowledge (Geracie et al, 2013).

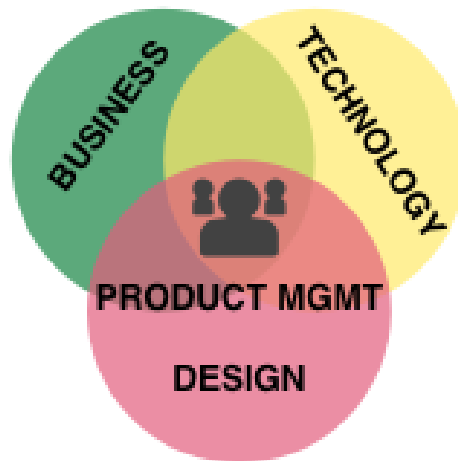


FIGURE 5 Cloud product management

Geracie et al. (2013) compare the software product management to manufacturing industry product management where the product manager must be able to support rapid growth, introduce new products based on new innovations, work with diverse and distributed manufacturing teams as well as supply chains with short development times. When this is combined with the cloud product definition the challenging domain of the cloud product management manifests itself. The cloud product manager doesn't need to be jack of all trades but she or he must rely also on the product team with needed substance knowledge of the different domains.

Business management including strategic thinking is the key to successful product management. Porter (2001) criticised already in 2001 in his article Strategy and Internet how strategic thinking and basic business modelling was neglected during the dotcom bust and it lead to the bust itself. Porter already updated back in 2001 his five competing forces model to the internet time and explained how the internet will change and impact all industries. This is even more prominent in the cloud product management today where new technologies, competitors and vendors emerge almost on daily basis.

The cloud product management is effectively about high technology product management and therefore it's crucial for the product manager to understand changes in the technology field. This can be seen from the Porter's (2001) five competing forces point of view or optimising the whole value chain with the latest technologies. In the chapter three there is provided an oversight for the cloud product manager how the cloud product value chain can be optimised from technology perspective, namely enterprise architecture and IT service management. In addition to that different and constant changes on application and infrastructure development tools and methods provide new ways to create value for the customer and drive innovation. These are con-

sidered to be part of the software engineering domain which is highly related to the cloud product management but is not analysed in details in this thesis.

The final domain of the cloud product management is the design which has two distinct focus areas: internal and external, like in the product management generally. The internal design domain is the design of business models, processes and capabilities which drive the product development. The external design is focused on the design of the product which includes the customer and user experience design.

2.3 Business modelling

Business model and modelling are according to Porter (2001) and Osterwalder (2004) highly debated and mostly vague concepts in business and strategic management research. There is no clear definition what is a business model and different schools of management provide different views to the business model. This was also one of the original reason why Osterwalder started his dissertation on the business model ontology and the business model design (Osterwalder, 2004).

According to Osterwalder (2004) the business model is an abstract conceptual model which represents the business and money earning logic of a company and provides linkage between the company strategy and processes. The business model role in the organisation is shown in the figure 6. Porter (2001) would argue that defining only earning logic is not enough but the business model should also take into account the industry structure through e.g. five competing forces and strategic focus should be on creating economic value not focus only on earning logic.

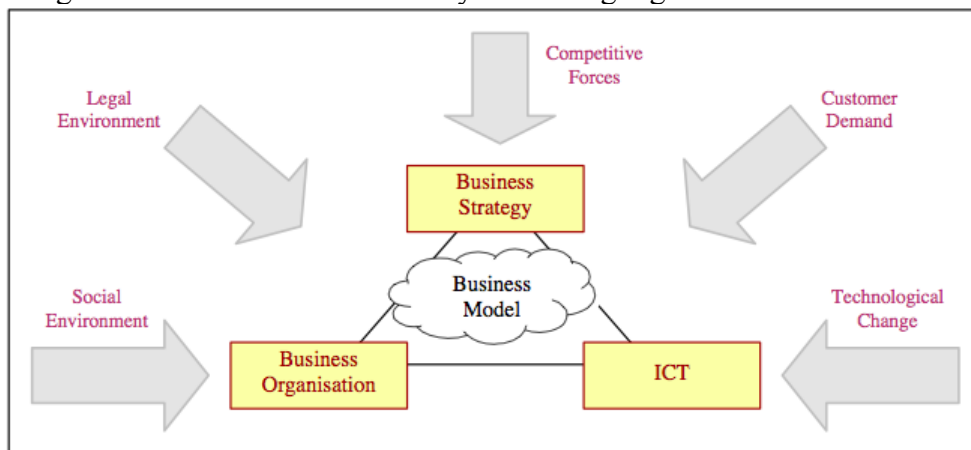


FIGURE 6 Business model role inside organisation (Osterwalder, 2004)

Even though Osterwalder's (2004) definition of the business model could have some missing attributes or definitions from strategy management point of view, it will be used through this thesis as the definition for the business model. Other aspects can be included into use of the business model, like Osterwalder has shown in his later work on the business model concept (Osterwalder and Pigneur, 2010). Osterwalder also acknowledges in his dissertation that most likely the definition wouldn't be accepted by all researcher. Osterwalder provides also ontology (layers or viewpoints) for different business views of the company where the top most level is the company business strategy

(planning level), in the middle is the company business model (architectural level) and the bottom most layer is the company process layer (implementation layer). The business model helps translating the company strategy to a blueprint how to implement the earning logic.

Osterwalder has continued his work on the business model ontology after the dissertation by simplifying the business model concept with the business model canvas (Business model canvas, 2015) and linking the canvas to strategic design and planning (Osterwalder et al, 2010). The business model canvas is simple yet powerful tool to describe on high level the company business model from value creation to target customers and it includes also cost and revenue streams. These can then be linked to actual implementation of the needed capabilities and the business model canvas is used in this thesis as a tool to describe the product vision. An example of the business model canvas and it's usage can be seen in the attachment 5. In order to understand the business model canvas and it's use the reader is referenced to Business model generation (Osterwalder et al, 2010).

The business model canvas (Osterwalder et al, 2010) can also be used to define new products and link them to the company strategy. Therefore the canvas is ideal tool for the product managers to showcase the product concept on high level. When following the business model ontology (Osterwalder, 2004) the product's business model could be used as the blueprint for the product including all the product aspects as defined in the previous chapters. Then this blueprint can be used to design needed capabilities of the product for implementation. This approach expands Osterwalder's (2004) process view (implementation level) to include all of the capabilities the product needs not only business processes.

The business model concept and canvas can easily be linked to the deliverables and tasks in the product management lifecycle defined by Geracie et al. (2013). Next the business model canvas (Osterwalder et al, 2010) nine building blocks are shortly introduced and relationship to the product management lifecycle is shown in the cloud product context. The deliverables in the product management lifecycle, like market investigation, should be understood by the cloud product manager and those should be implemented when they add value to the product development. These are not considered to be the deliverables of the product development process itself and the cloud product assessment method defined later in this thesis. The business model canvas with the needed deliverables from the product management lifecycle are considered to fulfil the additional documentation needed in the company in order to get commitment to the product development and are therefore more prerequisites for the product development than outcome of the product development.

Value propositions in the business model canvas (Osterwalder et al, 2010) should be used to define what value is delivered and which customer needs are satisfied with the offering. The supporting documentation from the product management lifecycle would be for example market & solution investigation, product charter and market feedback (Geracie et al, 2013). Value propositions is the most controversy to the cloud product definition and it doesn't take all product layers into account. This building block in the business model canvas should be considered to be used to define the product core benefits and how those are bundled and targeted to different customers segments.

Customer relationship in the business model canvas is used to manage customer expectations for the product. In the cloud product definition it was explained

that there might be customer segments which need for example access to the human resources and those models should be elaborated in the customer relationship with understanding of the cost model associated to such relationship model. These can be supported with market investigation and feedback documentation from the product management lifecycle. These same principles could be applied to the Channels and Customer segments sections in the business model canvas. It's important to recognise also other channels than the internet which is used to use the core assets of the product. The channels are related to sales, marketing, delivery and support models of the product and they should be integrated together to provide seamless customer experience for targeted customers. These sections could benefit also from the product and marketing strategies defined in the product management lifecycle. (Geracie et al, 2013.).

Key partners section in the business model canvas is used define partners, their role and linkage to the Key activities section. In the networked business models some customer segment can also be the key partner and executing key activities for the product and to the other customer segments (Osterwalder et al., 2010 and Hirsch et al., 2013). This can be seen for example in an online marketplaces where the customer can become a key partner by providing relevant content e.g. reviews for other customers. On the other hand the key partners can also be other (cloud) product vendors who provide needed resources or execute key activities. Key resources section is also linked to key partners and activities and is used to define the resources the product needs. These sections can be supported by wide variety of documentation defined in the product management lifecycle where product and marketing strategy, business case, marketing and deployment plan are the most relevant documents (Geracie et al, 2013).

Final two sections in the business model canvas are Cost structure and Revenue streams which cover financial aspects of the business model. Traditionally these are elaborated in a business case document through various investment calculation methods. One common problem with these investment methods is that they don't assess business risk related to the new product development. This can be partially assessed in various project management methods but the project risk management doesn't necessary cover all the risks related to the new product development, especially organisational development because typically project or program manager can't address these risks (Geracie et al, 2013). The lack of risk awareness and management is included into the cloud product capability assessment method in order to manage risks holistically during the product development. For various different investment calculation methods in the cloud business context the reader is referenced to Guide to the cloud computing (2013).

The cloud product cost structure and revenue streams are impacted by transformation in the cloud business from capital expenditure (CAPEX) to operational expenditure (OPEX) which is also implicit part of the cloud product value proposition. CAPEX means the cost of capital tied to IT investments which value is rapidly decreasing over time and they typically need large upfront investments. OPEX means operational costs which are variable costs tied to the business operations and scale according the business needs. (Hirsch et al, 2013). Shortly saying when the business scales up the OPEX costs increase and when the business scales in the OPEX costs decrease dynamically but in CAPEX model the costs are fixed and scaling happens in big leaps and it might not be possible to scale in the costs easily because of the upfront investment made in the past.

The change from CAPEX model to OPEX model might sound minor change but it's actually transformational change for many industries and especially for the cloud product industry. The change impacts all five competing forces like Porter (2001) showed already in 2001 and it impacts heavily to the cloud product business model. On the revenue streams the change impacts by customers having better negotiation position by getting more information and having easy access globally to many vendors. The customers are also expecting to buy cloud products with pay-as-you-go model which is related to the product scalability according the customer business needs. On the costs side the costs of suppliers also vary by the use if the cloud product is relying on other cloud products. It's also easy to take new suppliers into use when there is no big upfront investments needed. This increases innovation possibilities with the latest technologies but at the same time decreases the barriers of entry for competitors. Because the main benefits of the cloud products are delivered via the network access it also increases rivalry among existing competitors when all competitors have the no-cost or low-cost access to the channel itself.

Osterwalder (2004) also pointed that generally when people talk about business model innovation they are only talking about part of the business model like pricing or key partners who contribute to the value creation. In this thesis focus is on the new product development which is obviously a product innovation. Other possible innovation sources are for example process, marketing and organisational innovations. Typically these three innovation types are incremental innovations to existing products which impact some part of the business model like process innovation can decrease significantly delivery costs and therefore lead to better profit or access to new customer segments. Sometimes innovation can be disruptive on any of the aforementioned innovation types which means they impact whole business model and industry like online services selling music as subscriptions instead of consumers buying CDs. Therefore it's important to analyse innovation ideas from internal and external sources and understand their impact to the product business model. (Geracie et al, 2013.).

2.4 Business capability definition

The business capability has neither formal or clear definition like is the situation with the business model definition. TOGAF enterprise architecture framework (Open Group Standard, 2011) defines business architecture as an architecture area which covers business strategy, business processes and governance inside the company. In the business model definition Osterwalder (2004) explains there is multiple views on business which can be layered to planning, architectural and implementation layers (see chapter 2.3). TOGAF's definition covers all Osterwalder's business view layers.

The target of this research is to build an assessment method for cloud product development. The business modelling is already defined in the chapter 2.3 to be an abstract layer to map the company strategy to a company blueprint in order to implement the strategy. In this thesis the business model is expanded for the product life cycle and development purposes to manage all the capabilities the cloud product needs. The capabilities map to the implementation layer of the business in the Osterwalder business model ontology (2004) and in this thesis the capabilities are the implementation layer of the cloud product. The actual implementation and delivery of the capabilities is not

covered in the business model concept and neither in this thesis. The capabilities are classified to different domains in order to manage them in the cloud product capability assessment method, see chapter 4.1, and the business capabilities are considered to be only the business processes the product needs which are managed during the product development and life cycle.

According to Ko, Lee, S. and Lee, E. (2009) there is also a lot of work to do in the business process management research to define what is a business process even though over three decades is already spent on analysing and structuring the business process management with different frameworks, tools and languages. A lot of effort is invested in building different ways to manage and describe the business processes. This thesis won't address the business process and the business process management discussion but gives a simple definition what is the business process from the cloud product and it's life cycle point of view.

The product management has the focus on managing products inside the company and representing the customer in the organisation. The cloud product definition and the business modelling also explain all product layers and why they are important for the customer and the product concept. Based on these the cloud product business capability definition is:

- process interferes directly with the company external parties, including customers and partners, and generate revenue or are part of the value proposition of the product
- process assess one or multiple layers of the cloud product
- process typically overlaps multiple organisation units

Implicit to this definition is that there can be many other product related processes which could be considered to be business processes by definition. These processes will then be part of other product capability domains, see chapter 4.1.3 for detailed information. The business capabilities focus is on adding value for the product or generating revenue which then directly impact the product business model.

Even though the actual implementation of the business capabilities or the business process modelling is not managed in this thesis it's worth noticing the importance of the modelling. The business capabilities should be modelled the same way than other business processes in the organisation in order to maintain coherence of the models used inside the organisation. Even though the product manager wouldn't be designing the business processes it's important to understand the purpose of modelling and how the business processes could be modelled. The reader is referenced to Enterprise Architecture at Work. Modelling, Communication and Analysis (Lankhorst, 2012b) to get overview of modelling practices and importance of the models as a communication mechanism.

2.5 Business agility is needed

A lot is written about agility in the context of business and IT management. Agile practices emerged from the Agile manifesto (Manifesto for Agile Software Development, 2001) in the software development context and has afterwards expanded to other areas

like general IT and business management (Anderson, 2012 and Lankhorst, 2012a). Even though many of the practices have existed in lean manufacturing industry long before the agile buzz began in the software development communities, see for example *Out of the crisis* (Deming, 2000).

This thesis won't address all different agile development practices but provides short introduction what agility means in the context of the cloud product management and business and why it's especially important in the cloud product industry. These findings are generally applicable for the whole software business industry including for example mobile applications and on-premise software but those are not covered in details because each of the domains have own characteristics. Some of the reasons why the cloud product management needs agility is covered already in the business modelling, see chapter 2.2, including the transformational role of the internet and related changes in the cost structures.

Before business agility can be addressed it must be defined. In the software development context the agility is typically defined through the Agile manifesto (2001) and how well different methods comply to the defined principles in the Agile manifesto. Focus with agile development methods typically is on how effectively, sustainable and with high quality the software can be developed. This is not enough for the product development because agility of the product should be operationalised. (Lankhorst, 2012a.).

Operationalising means that the product should be able to deliver it's value through it's life cycle. For example business capabilities are heavily influenced by the people participating in value creation as explained in the previous chapters. On the other hand developing such capabilities can be called as a wicked problem. The wicked problem in the cloud product development context can be summarised as an unique problem which can't be understood before a solution is built, there is no right or wrong solution to the problem and there is only one solution at a time to the problem and after implementation of the solution the original problem is altered. Social complexity impacts wicked problem implementation and people in the organisation can block or seriously interfere with the solution implementation if they don't accept it. (Lankhorst, 2012a.).

Based on the complexities of the service development Lankhorst (2012a) has built a definition for a agile service which can be expanded to the cloud product context: "An agile cloud product is a product that can accommodate expected and unexpected changes rapidly in order to maintain and increase it's value proposition in sustainable way". The definition for the agile cloud product takes also Agile manifesto values into account by incorporating sustainability which should be considered through all the product capabilities and take people, processes and technologies development into account.

Next the agile cloud product definition is compared to the seven common business agility drivers Lankhorst (2012a) has found. *Product dynamics* is the demand from customers and market as well as new business opportunities. In the business modelling chapter, see chapter 2.3, internet role is defined how it has changed the demand and supply for the cloud products. In addition to that major influence for the demand comes also from industry clock-speed which means how fast the industry is changing. The need for rapid change can even be seen on traditional ERP suites which are looking new ways to incorporate and integrate ERP solutions to the cloud world or even use more and more cloud based solutions (Gartner, 2014b). Because the cloud product in-

dustry is moving relatively fast customers are also expecting to get better and better value out of the products compared to the old upfront investments to onpremise applications and the cloud provider must address this demand by continuously updating and altering their offering.

Revenue dynamics is according to Lankhorst's (2012a) demand for changing pricing strategies which influence to the revenue streams. The cloud product company must be agile and adapt to this all the time changing situation which means major changes for example company budget forecasting and cash management. *Volume dynamics* means that demand and supply can vary. The cloud product company must be able to adapt to these changes on continuous basis which means scaling the resources needed dynamically. Scaling can be hard if there is a lot of product capabilities which need human resources.

Channel flexibility is the business need to change rapidly used channels as defined in the business model. This demands new channel management practices for the cloud product companies, especially if channels are adjusted dynamically which can be the case for example in dynamic consumer markets. *Supply chain flexibility* is the need to adjust the supply chain dynamically based on the demand. In the cloud product business this means scaling the supply chain and supply usage based on volume dynamics e.g. active users and adapt computing resources accordingly. *Continuous compliance* is the business need to adapt any changes in the legislations and other compliance needs which can come also from customers. For the cloud products this is highly important when already basic legislation for example personal information data and storing of it varies significantly between countries. The global nature of the cloud business and increasing customer demand to fulfil the known compliance frameworks for example in security, finance or personal information data management needs agility from the vendor to adapt to these constantly changing requirements.

Final business agility driver is *technology adoption* which means adopting new technologies in order to drive cost efficiency or gain competitive advantage over competitors. Because the cloud products are relying other technology vendors on supply and partner side it's important to understand the agility needs for the vendor management or even having multiple vendors to provide same capabilities in order to have less vendor lock-in. In traditional vendor management principles multiple vendors are considered as a risk because of tightly coupling and alignment of the vendors. The cloud vendor management can differ significantly from this approach because the consumer (the cloud product vendor) can manage needed resources mostly automatically and without vendor involvement as explained in the cloud product definition.

Above examples of the cloud product business and the research findings shows why the business agility is needed in the cloud product management. Agility is achieved through continuous development and maintenance of different capabilities over the product life cycle. The reader is referenced to Agile Service Development. Combining Adaptive Methods and Flexible Solutions (Lankhorst, 2012a) to get more information how such agility could be implemented.

3 IT MANAGEMENT VIEWPOINT TO CLOUD PRODUCTS

The IT management viewpoint to cloud products is a short introduction of the IT service management (ITSM) and enterprise architecture (EA) frameworks and their relation to the cloud product. Target with this chapter is to give a basic understanding of the selected IT management frameworks for product managers and other people working with the cloud products. Each of the frameworks have comprehensive set of practices and knowledge included which can't be discussed in details in this thesis. These frameworks will be analysed on more detailed level during the cloud product assessment method design and ontological analysis, see chapter 4.1.4.

The common nominator from business point of view for ITSM & EA frameworks, methods and research is IT and business alignment. This demand to align and build new practices for the IT and business alignment raised in the end of 80s and beginning of 90s when IT role changed from administrative support to more strategic enabler inside organisations (Lankhorst, 2012b). Henderson et al. (1993) made their well-known research on IT and business alignment over two decades ago. In their research they showed how business strategy can be linked to IT strategy as well as how business process management is related to IT process management. Their model provided different viewpoints to the organisation and how IT can be used as strategy enabler or how IT strategy can drive the company strategy.

IT and business alignment is also studied in vast number of different research papers but simple solution is not found how to implement it. Shpilberg, Berez, Puryear and Shah (2007) even question in their research if this alignment is even the key success factor. They found that companies where IT and business were highly aligned the companies spent 13% more on IT and generated 14% less revenue compared to the averages. They call this to “alignment trap”, a situation where more IT resources are spent on alignment with poor results. At the same time they found companies which spent less than average on IT but generated 35% more revenue. They called these companies to “it-enabled growth” companies.

The key finding of these studies is that maybe the alignment is not what companies need but better business results. The cloud product business is heavily relying on IT so alignment should be incorporated to the product development not thought

as separate activity. The cloud product manager has important role on bringing the IT and business together like explained in the chapter 2.2.

Another important study on enterprise architecture is done by Ross (2005) regarding organisation operational model. In the research Ross explains four different operational models for integrating and standardising business processes in the organisation. The levels vary between high standardisation and high integration to low standardisation and low integration. In the cloud product management it's important to understand which is the operating model in the organisation in order to get business capabilities implemented accordingly. For example in a high standardisation and low integration operational model the business capabilities must be implemented and maintained with each business unit because there is no coordination and cooperation between the units. This means more work during the product development.

As the above examples show the IT viewpoint and research can provide valuable input to the cloud product development. Next four common frameworks are shortly introduced in the context of the cloud product management. The reader is referenced to Enterprise Architecture at Work. Modelling, Communication and Analysis (Lankhorst, 2012b) for introduction of the other possible frameworks.

3.1 IT Service Management frameworks

3.1.1 CobiT 4.1

CobiT is an IT service framework developed by IT Governance Institute. CobiT 4.1 provides reference process model and common language for organisations to manage their IT activities. CobiT target is to help organisations secure business benefits from IT investments, IT and business alignment and manage corporate risks related to IT. CobiT is focused on IT governance not on managing any process implementation in details. CobiT is also aligned with enterprise architecture. Figure 7 illustrates this alignment on process and systems level. (IT Governance Institute, 2007.).

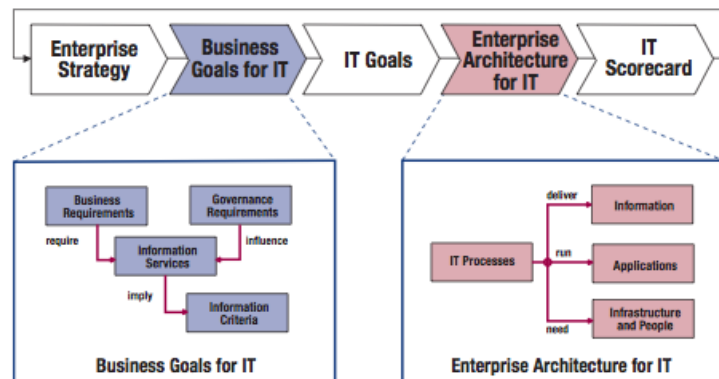


FIGURE 7 CobiT alignment to enterprise architecture and business (IT Governance Institute, 2007).

Key elements in CobiT framework are process maturity model, process goal and measurement model and methods how to do process governance. The framework is divided into four different process domains and 34 different process descriptions from governance and reference point of view. Process domains are Plan and Organize, Acquire and Implement, Deliver and Support, and Monitor and Evaluate. (IT Governance Institute, 2007.).

Figure 8 illustrates CobiT process model ontology and elements relations to each others. From capability point of view the ontology model clarifies significantly how reference processes should be managed and what are most important assets. Another important ontology in framework is process maturity model which is split to six levels. This ontology is based on Software Engineering Institute Capability Maturity Model (CMM) which is well known on IT domain and not discussed in details in this thesis. (IT Governance Institute, 2007.).

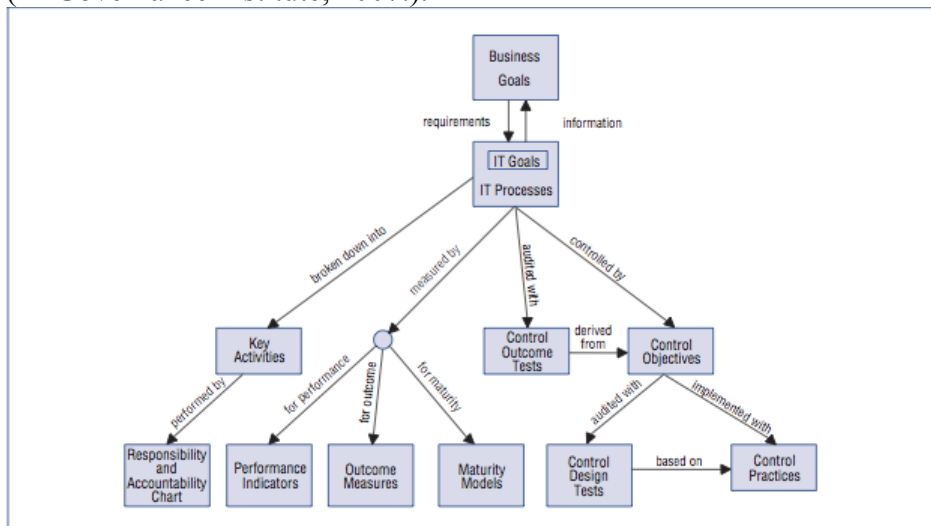


FIGURE 8 CobiT process model ontology (IT Governance Institute, 2007).

For the cloud product management CobiT provides clear governance structure and mechanisms which can be taken into use if there is customer or other like legislation need for well structured governance model. Even if the framework wouldn't be implemented thoroughly it provides well defined control objectives for example business capabilities.

3.1.2 ITIL v3

IT Infrastructure Library (ITIL) is another well known IT service management framework. According to Nabiollahi et al. (2010) ITIL has transformed significantly between versions 2 and 3. Version 2 concentrated more on IT service delivery and support when version 3 takes the whole service life cycle into account (Nabiollahi et al., 2010). This research evaluates ITIL v3.

According to Macfarlane and Taylor (2007) ITIL is the best practice for IT service management but should not be the only practice the organisation follows. Their implementation guideline suggests taking holistic approach to IT service management

and link ITIL best practices to other well known IT management practices like TOGAF, CMMI, PMBOK, PRINCE2, SOA and COBIT. It is clear from the implementation guidelines that proposed frameworks are essential to manage the service life cycle in the organisation but not much guideline is given how to align the practices (Macfarlane et al., 2007). From capabilities point of view ITIL concentrates more on process management best practices not aligning the framework with other frameworks like CobiT is doing.

ITIL is divided into five different process domains and 25 different processes targeted to manage the service life cycle (Nabiollahi et al., 2010). The service life cycle is the key concept in ITIL when evaluated it from capability point of view. Capability management itself is an incremental process in ITIL. (Macfarlane et al., 2007).

According to Macfarlane et al. (2007) key principle in ITIL is how business value is created with services by measurable outcomes. In the core of ITIL is the service strategy process domain which defines the service management as an organisational capability and strategic asset. (Nabiollahi et al., 2010). Next layer in ITIL is based on the service design, service transition and service operations process domains which are all linked back to the service strategy. Uppermost layer in ITIL is the continual service improvement which provides governance layer over all above process domains. (Macfarlane et al., 2007.).

The service design domain provides guidance for designing and developing services as well as the service management principles. This domain is also linked to technology and architecture management. The service transition domain provides guidance how the service design is turned into capabilities in the organisation. The service operation provides guidance how to manage services and provide essential information for other service life cycle parts. It's claimed to be process domain where vast majority of IT budget is spent. (Macfarlane et al., 2007.).

From the cloud product management point of view ITIL can be seen important framework because it provides common language for the enterprise cloud product market and more detailed implementation guidelines than CobiT. ITIL provides also almost as strict governance and segregation of duties framework as CobiT. All the processes and practices are developed for internal IT management purposes. This is also the Achilles heel of ITIL when comparing it from to the cloud product point of view.

Customers are expecting the cloud product vendors to be compatible with ITIL practices what they are internally using. At the same time the definition of the cloud product demands more responsibility and authority for the customer of the resource usage which provides loosely coupling between the vendor and the customer. This provides also vendors the possibility to scale out the offering and provide the cloud products for broader audience cost efficiently. ITIL doesn't assess yet well the cloud products as part of the internal application portfolio which can cause conflicting interests between the vendor and the customer.

3.2 Enterprise Architecture frameworks

3.2.1 TOGAF 9.1

TOGAF is full scale enterprise architecture framework which is publicly available for everybody interested in the enterprise architecture. TOGAF is developed since mid 90's by The Open Group and latest version is TOGAF 9.1. Framework is divided into seven different parts. (Open Group Standard, 2011.). Key concepts and models are introduced from the capability point of view.

Architecture Development Method (ADM) is one of the key concepts in TOGAF. Figure 9 illustrates ADM and different architecture domains linkage to enterprise architecture requirements management. ADM is considered to be incremental process to manage and develop enterprise architecture. Each of ADM sections are described in TOGAF with details which tangible and intangible assets organisation must manage to have enterprise architecture capabilities in place. (Open Group Standard, 2011.).

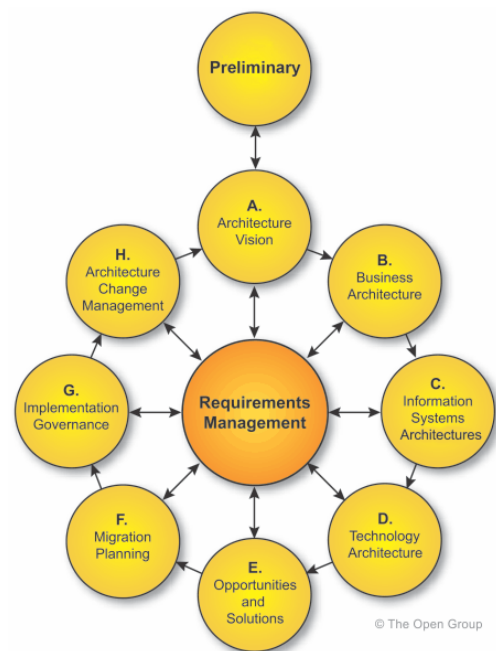


FIGURE 9 Architecture Development Method (Open Group Standard, 2011).

According to other research of capabilities management and IT service management, see e.g. Nabiollahi et al (2010) and Correia et al. (2009), the four TOGAF domains are considered to be most relevant to this thesis. TOGAF business architecture domain objectives are to develop target business architecture which includes products and services strategy, organisational, functional, process and geographic aspects of business environment. (Open Group Standard, 2011.).

Next step in ADM is information systems architecture domain which defines data and application architectures. Data architecture addresses generally how data is managed in the organisation context and how it enables business architecture. Application architecture defines how application components are created and managed in the organisation through their life cycle to enable business architecture. Technology architecture defines technology building blocks from logical to physical components that enable the business architecture. (Open Group Standard, 2011.).

For the cloud product management TOGAF framework provides holistic approach how the architecture of all product aspects can be managed. TOGAF is not limited to one view but provide four different views to the architecture. The cloud product manager is not expected to deeply understand the architecture framework because it belongs to architecture management domain but the key concepts are simple yet powerful ways to describe the cloud products. An example of utilising and integrating TOGAF in the domain of product management can be seen in the development of the cloud product capability assessment method, see chapter 4 for details.

3.2.2 Zachman v3

Zachman (1987) has originally defined how different views and the ontology of Zachman framework is developed from the need to communicate enterprise architecture and business requirements to different stakeholders. Figure 10 shows the latest (3.) version of Zachman framework for the enterprise architecture management. Enterprise architecture is considered to be an ontology of different assets that enterprise architect must manage to build up enterprise and needed information systems architecture. (Zachman, 1987.).

Zachman framework has six different viewpoints to the architecture. Ontology defines how different viewpoints are communicated to different audience. In addition to that ontology points out each entity relationship to upper and lower viewpoints as well as other entities at the same viewpoint or layer. (Zachman, 2011.). Zachman (2000) has clarified in his article that different viewpoints are essential for building up enterprise architecture. He classifies architecture viewpoints to three different categories: conceptual, logical and physical which are relative to the current viewpoint. The current viewpoint or architectural layer is always the logical viewpoint, the above layer from the current viewpoint is the conceptual viewpoint and the below one is the physical viewpoint.

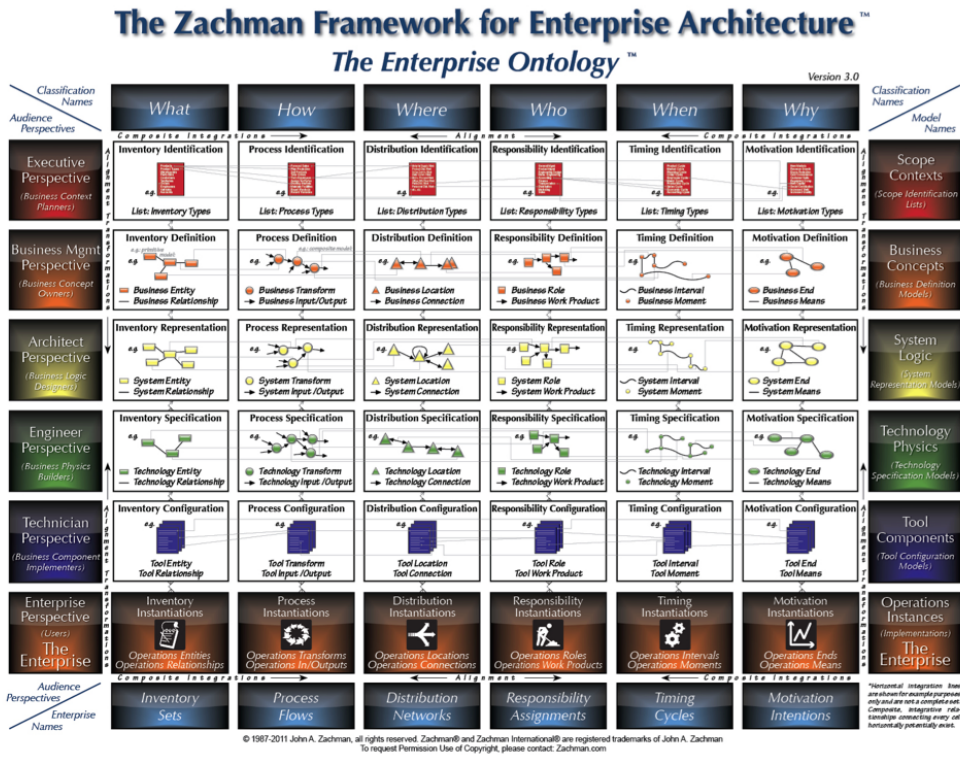


FIGURE 10 Zachman framework v3 (Zachman, 2011)

Zachman framework is introduced to give more a historical point of view to the enterprise architecture management. For the cloud product management Zachman framework layered viewpoints can provide simple tool to address different stakeholder communication needs. For example the business model description can be enough details for senior management and therefore considered to be logical view to the product but for the product team the same presentation could be conceptual view of the product. It is important to understand different stakeholder needs for details.

4 METHOD DEVELOPMENT

4.1 Method design

4.1.1 Cloud Product Capability Index

Cloud Product Capability Index (CPCI) and its domain model is shown in figure 11. CPCI is relative not an absolute index for the different capabilities the product needs. Cohn made this approach famous in his *Agile planning and estimating* (2005) book. In his book he explains why traditional estimating and planning fails. Mostly it's because planning is hard without all the information and knowledge upfront and the target is also changing when new information is found. This uncertainty leads plans to fail so often.

If plans and estimates fail so often should planning even be done? Some agile approaches propose that there should be no planning at all. Cohn (2005) explains that there is typically many legitimate reasons why planning should still be executed. According to Cohn these legitimate reasons are typically related to dependencies to other teams and activities. Dependencies to other teams is the main reason why the cloud product capability assessment has a planning mechanism. The product development is cross-functional activity creating demand through product life cycle to the whole organisation and it should be managed.

Cohn (2005) describes good planning process characteristics and the cloud product capability assessment method should support all of them. CPCI should highlight capability development risks by taking into account group assessment and weighting it with business risk. Development uncertainty should be decreased with structured way to define product business model, needed capabilities and continuously assess them with the latest knowledge. Decision making is supported by making development status visible all the time, showing the work not done and what is already achieved. This can help making formal scope, budget and timing decisions.

Cohn (2005) and Anderson (2012) both highlight that trust and driving fear out of the teamwork are essential characteristics of agile environment. Continuous

collaboration, open discussion of business model, capabilities and risks should create an environment where each team member feel empowered and willing to contribute on analysis. This should lead to better quality and more reliable estimates. Plans should also convey information and stakeholders should be able to draw conclusions from the plan by taking a look on progress as much as assumptions and limitations defined in the assessment. Relative assessment means the results are not comparable between teams or products and that is not the intention of the method.

CPCI mathematical representation can also be seen in the figure 11. Individual cloud product capability is linked to product business model. Each product must have at least one business model but it can have multiple business models like in platform business.

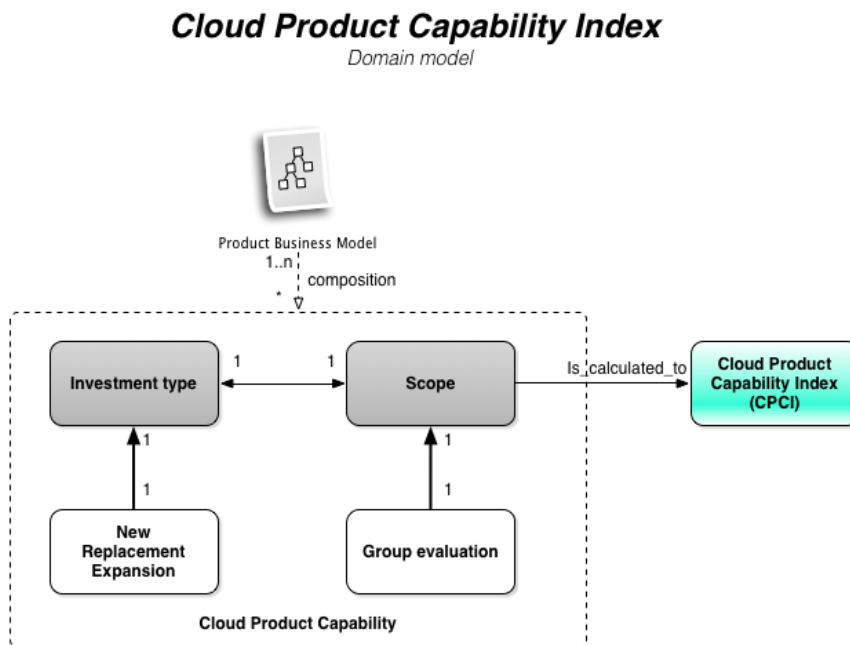


FIGURE 11 Cloud Product Capability Index domain model

Individual product capability has two factors: investment type and scope. Investment type has three properties and each property has own factor. Investment type reflects and classifies the business risk where new investment means that the capability is totally new for the organisation and it has the highest risk. Replacement type means that the organisation has already such capability for example in other product and the capability will be replaced or replicated to the product under assessment. Replacement investment has the smallest business risk because organisation has already the needed knowledge. Expansion investment means that existing capability will be expanded to a new area e.g. new customer segment, with a new partner or capability is incremental improvement to an existing capability. Expansion risk level is between with the new and replacement investments. Business risk has the following base factors: new = 2, expansion = 1,5 and replacement = 1.

Product capability scope is estimated in group assessments and evaluation follows Cohn (2005) relative size assessment. Size represents overall size of the capability not the duration how long it takes to deliver it. In this thesis following growing scale numbers are used to estimate capability size: 1,2,3,5,8,13,20 and 40. Basically any other quantifiable scale could also be used. Cohn explains that the relative scope size assessment includes multiple aspects like amount of effort needed, complexity and risk. Relativeness comes from the Cohn assessment method; either team selects rather simple capability and gives it the smallest number or medium complex capability is selected and it gets number somewhere in the middle of the scale. After that all other capabilities are compared and evaluated against this “baseline” capability.

Finally CPCI is calculated simply by multiplying the investment type with the scope. For example a capability which investment type is expansion and scope is 8 CPCI is calculated as: $CPCI = 1,5 * 8 = 12$. Each capability can have only one investment type and scope at a time but these factors can be updated when needed.

4.1.2 Capability development framework

Getting a product from inception to market is complex manoeuvre like it's elaborated in the chapter 2.2. The product development can have multiple phases, stakeholders and the target can be changing over time. Nevertheless the product development is typically considered to be quite linear or sequential process from inception to go-to market. Also enterprise architecture management is typically summarised to be defining the current situation (AS-IS), the target situation (TO-BE) and the gap between them, see for example Guide to cloud computing (Hill et al, 2013) or TOGAF Migration planning (Open Group Standard, 2011).

These linear approaches to complex development topics like the product development or enterprise architecture are needed in order to communicate context and situation to many stakeholders. Capability development framework also provides common terminology and language for others to understand discussion, context and compare different cases with each others. Without common language everybody should have the same information and in-depth understanding of the product and development status than the product team which is impossible to achieve. Without common language and development framework discussions might also lead to misconceptions or misunderstanding when people could be using different terms to have the same meaning or the same term to have different meaning. Therefore the product team needs ways to communicate the product development status to the stakeholders. This capability development framework is shown in the figure 12.

The capability development framework visualisation follows classical approach where target situation is defined as to-be situation with all of the capabilities needed in the go-to market situation. The baseline are the existing capabilities in the organisation. The development framework has also traditional gap analysis as well as linear progress of development visualised in order people to easily recognise and understand the capability development framework.

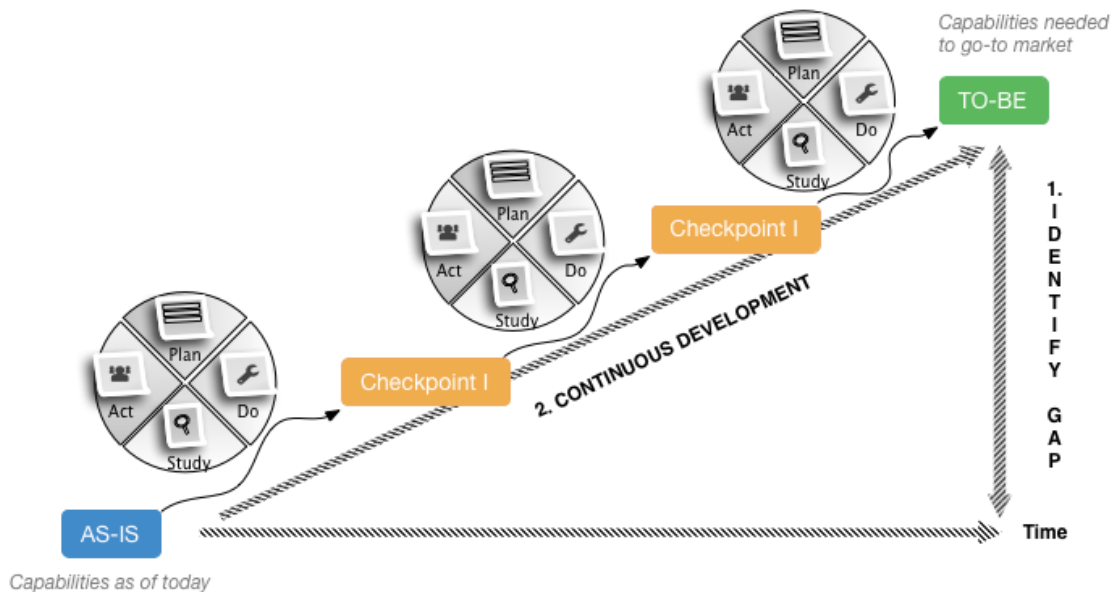


FIGURE 12 Capability development framework

The major difference between sequential or linear development process and the capability development framework is the adaptability. The development framework should adapt to any changes whether they are context, scope or time. This is needed in order to support agility as well as constant changes in the organisation as explained in the chapter 2.1. One agile attribute of the framework is that it should be easy to map with any existing product life cycle models used in the organisation. This has also been common way with agile software development methods and how those are taken into use in organisations which are using more waterfall type development processes, see for example Anderson's Lessons in Agile management (2012).

The capability development framework core part is the continuous development process which is described in details in the next chapter. The starting point of the continuous development is the current situation which is defined as AS-IS situation. This means that product manager and the product team should recognise that there is typically many existing capabilities in the organisation which needs to be changed, modified or tailored to the product needs, see investment types in the previous chapter. Basically this means that development rarely starts from *tabula rasa* situation unless there is a new company or business unit created to take care of all the capabilities.

There is no AS-IS situation analysis or capability assessment like TOGAF (2011) propose included in the AS-IS phase. The reason is that such analysis is included into the continuous development model. Capabilities status in the organisation is not static and therefore too much emphasis and effort shouldn't be allocated to the current situation analysis. This of course can be done if the product manager or the product team is new to the organisation but it's considered to happen before the product development starts or it's happening during the development as part of the continuous development process.

TO-BE situation is defined target for the product before it can be launched to the market. Like in AS-IS phase there is no formal gate or criteria defined when this

situation is achieved because it varies much between the products. Needed capabilities can be anything from so called Minimum Viable Product (MVP) concept where only bare minimum capabilities are built or something complex like full scale certification needed in health care industry. It is important that the product team knows the target situation in the go-to market and these restrictions and limitations must be elaborated as part of the individual capability assessment.

The defined target for TO-BE situation is considered to be the output of the continuous development process and capability assessment. All the capabilities defined as part of the assessment are the capabilities needed in TO-BE situation. If they are not needed they should be scoped out of the current assessment and TO-BE situation. TO-BE situation is also clarified by utilising business model canvas (Osterwalder et al., 2010) which shows the product vision current status.

The framework visualisation (figure 12) also includes gap analysis as a visual element with two headed arrow. This represents classical gap analysis between AS-IS and TO-BE situations. The reason it has two headed arrow is that it's considered to be continuous activity not one off activity. Product development starts with the first definition of the TO-BE situation and then the gap analysis can be executed. Basically this means the first analysis of the needed capabilities which should be compared to the capabilities defined in the assessment template. Analysis means also the capabilities defined in the assessment template should be removed if they are not needed in the TO-BE situation. The assessment template provides only the most common capabilities from enterprise architecture and IT service management frameworks which might be relevant for the new cloud product.

Following list shows the execution order of the capability development framework tasks:

1. Understand AS-IS situation
2. Define TO-BE situation with the business model canvas and the assessment template in the cloud product assessment tool
3. Analyse the gap between AS-IS and TO-BE capabilities defined in the cloud product assessment tool
4. Execute the continuous development process

The continuous development process has two important visual elements in the framework. First of all, continuous development route is visualised as curved arrow from checkpoint to another. This means that the route towards the next checkpoint or TO-BE situation is not known or defined upfront but it's continuously corrected towards the target situation. This follows agile principles as well as cycle of continuous development defined by Deming (Anderson, 2012).

Another important aspect in the framework are the checkpoints. The continuous development process defines that the process is executed with regular interval. Each process execution creates one checkpoint of the continuous development process. These checkpoints can also be mapped with any predefined product life cycle process phases in the organisation. This way the continuous development can follow for example defined sequential product life cycle but between the phases process can be executed in more agile manners.

During the initial gap analysis and the continuous development process there can be many capabilities identified which are not necessary needed in the go-to market situation. Typically in agile development methods it's also relevant to define what will not be developed. In the same manner it's important to keep list of the capabilities which are not developed for the go-to market situation. This list works effectively as a product backlog for incremental product development. Incremental product development is not part this thesis but the framework should be applicable for product incremental development after the go-to market situation.

4.1.3 Continuous development process

The core part of the capability development framework is the continuous development process. The continuous development process focuses on the product development not operational implementation of an individual capability. This is important distinction to made between operational development methods like Scrum or waterfall and product development. The continuous development process is shown in the figure 13.

In the previous chapter the capability development framework is explained in details. The input for the continuous development process is considered to be the initial steps of the capability development framework. It is shown as Inception in the continuous development process. Output of the Inception is the initial draft of the product, needed capabilities and the first gap analysis.



FIGURE 13 Continuous development process

The aim with the continuous development process is to align with Deming (Deming, 2000) famous management principles and focus on systems thinking instead of quality control. Deming (2000) discusses in the book how important it is to think customer as the most important part of the production line and how management should focus on

quality of the process instead of conformance to specifications. Anderson (2012) has also shown how well Deming management principles map to agile principles. Quality is everybody's responsibility and organisation should be seen as complex living organism which needs to be optimised as whole instead of focusing on individual parts.

The major difference when building a new product and incrementally improving existing product is the lack of customer. Even though there is numerous ways to do design from customer perspective, involve prospective customers into design and evaluation of the new product with them the real (paying) customer is still missing. After getting the product to the market the real customers begin to be infinite source of internal and external demand for the product development. This demand management and incremental product development is out of the scope of thesis. Getting the product to the market therefore relies on the ideal description of the customer and the capabilities the customer is expected to need.

The continuous development process has four main phases: *Plan*, *Do*, *Study* and *Act*. The process is considered to be incremental and each process execution produces a new checkpoint of the product development. As explained earlier checkpoints can be mapped with any existing milestones organisation uses for product development. Existing milestones in the organisation could be considered only the next target situation not the increments the continuous development process should produce. It is important to agree defined cadence (process cycle time) for the continuous development process because it will be the basis for the product development. The process cadence could be for example two weeks which means that all process steps are executed during that time period.

The plan phase takes input from the inception or the previous act phase. In the plan phase the product team works together by updating the product vision and executing assessment of individual capabilities. Output of the plan phase is updated version of the cloud product capability assessment which defines needed capabilities in priority order, see figure 14 for an example situation. In the cloud product capability assessment capabilities are classified according to TOGAF (Open Group Standard, 2011) domains which was found to be the most relevant classification for the method, see chapter 3.2.1 for additional information.

There is couple of deviations for the capability assessment compared to classical EA viewpoint. Instead of planning capabilities from the whole organisation point of view capabilities are planned from the product point of view and compared what is product impact to existing capabilities as suggested in the capability development framework. Another clear difference to EA and general architecture principles is that target is not to support composition and integration of different domains but provide a way to have common classification and terminology to discuss of different capabilities. According to Lankhorst (2012b) this would be called a product viewpoint. Other viewpoints can be elaborated as part of the implementation of individual capability but those are not in the scope of this thesis.

Next the product viewpoint is compared to TOGAF (2011) domains and how individual capabilities should be analysed from the product viewpoint. The product viewpoint assumes the product manager is interested in understanding and defining capabilities and their requirements from the customer point of view not how they should be implemented.

According to TOGAF business architecture is prerequisite for architecture work in any other domain. Such strict limitation is not needed in the product viewpoint because products are developed into organisation which has already some business capabilities in place. Therefore business capabilities should focus on defining needed business processes and how to adapt the product to the existing capabilities. Needed business processes should be derived from the product business model which is defined by using the business model canvas (Osterwalder et al., 2010).

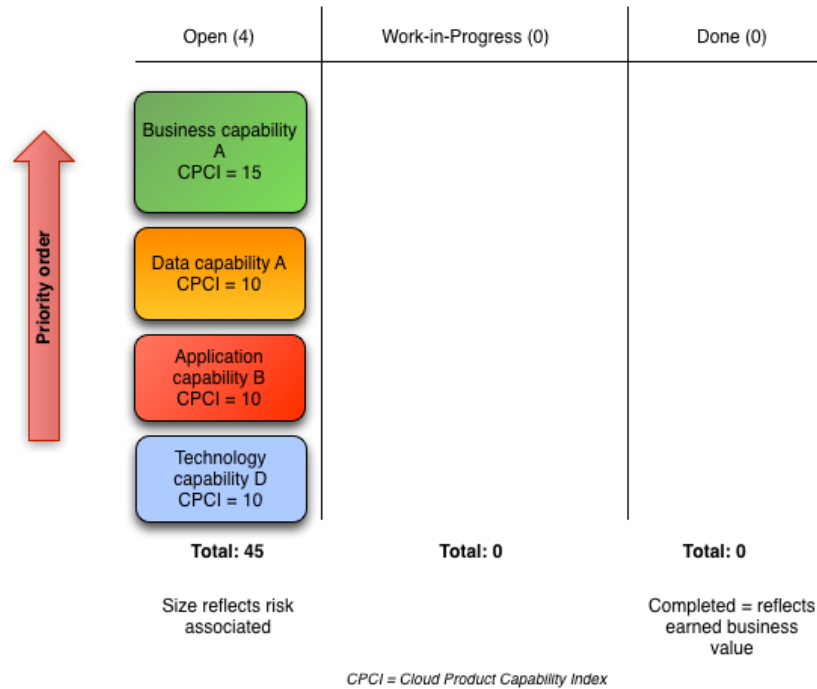


FIGURE 14 Continuous development process in the first execution

TOGAF (2011) defines data architecture and data management as crucial enabler of architectural transformation. From the product point of view there is a lot of important data which customer needs or organisation needs of the product in order to execute business processes. These capabilities are classified as data capabilities.

The cloud product is described earlier to be high technology product which consists of different kind of application and technology components which are related to the product cloud model e.g. IaaS vs SaaS. Therefore much emphasises from the product viewpoint is added on planning and defining these capabilities. The application capabilities include capabilities, typically called features, needed from customer and business processes point of view. That way for example integrations to existing backend systems are taken into consideration early in the planning phase. These integrations enable business process execution by transforming and integrating needed data between the processes.

Like in TOGAF technology architecture also in the product viewpoint technology components are considered to be the enablers of a product. In the product viewpoint technology components are defined from customer point of view for example what is demanded or needed from a vendor or how much capacity is needed. It's also important to realise if product target technology capabilities are aligned with organisa-

tion guidelines and competencies in order to address risks associated with different or possible new technologies in later product life cycle phases.

In the next chapters ontological analysis is conducted between EA and ITSM methods in order to build a template of the needed capabilities for the cloud product. This template is enriched later on with other sources to build comprehensive list of capabilities which should be taken into consideration in the plan phase. These are not meant to be mandatory capabilities for all cloud products but as a starting point of a capability assessment.

Each product capability has also other attributes than the domain they represent. Capability has name and description and description should be written from the product viewpoint according to each domains. Capability has also CPCI index which is calculated from the parameters defined in the chapter 4.1.1. Capability has status parameter which is used to follow the progress. Statuses for a capability are *Open* when the capability development hasn't started, *WIP* (work-in-progress) when the capability is under development and *Done* when the capability is done.

From the capability development point of view each capability has also following attributes: definition of done, priority and owner. Definition of done is agreed together with the product team when the capability is ready and the definition should be as unambiguous as possible. Definition of done should be used to record limitations and assumptions of the capability implementation. Priority shows each capability development priority compared to other needed capabilities. Owner is the person in the product team who is responsible of the capability implementation. All the capability attributes can be updated in the plan phase.

The next phase in the continuous development process is the do phase. In this phase individual capabilities are implemented according the definitions and limitations updated in the previous phase. Implementation or operational delivery of the capabilities are not covered in this thesis. Implementation starts from the most important capability in priority order. When implementation starts capability status is updated from open to WIP.

Individual capability life cycle follows this simple Kanban flow from open to done. Kanban development method for software development was made famous by Anderson (2010) in his book “Kanban: succesful evolutionary change for your technology business”. Like Anderson explains in the book such flow itself is not yet Kanban system but it needs many other aspects like limiting work in progress and statistical analysis of the process. Simple flow should be called only visual control system.

For the clarity reasons in this thesis development flow is referred as Kanban flow and the development status as Kanban board because the capability development framework has many other Kanban characteristics like focus on discussing and optimising product development together, collaborative assessment and visual signalling. Statistical control of the process is scoped out because there can be different teams implementing individual capabilities. And development is not considered to be one continuous development flow but more of parallel development streams.

The next phase in the continuous development process is study. This phase has two main targets: analyse progress of the development and execute the gap analysis. In the figure 15 an example situation of a product development is shown. Even though this visual signalling system is simple it provides a lot of information about product de-

velopment. On the left side of the Kanban board development backlog size can be seen which reflects how much work still needs to be done.

In the middle of the Kanban board can be seen amount of unfinished work and on the right side completed work. Completed work can be called earned or potential business value which will be realised when the product is ready to the go-to market. Each individual capability is also shown on the Kanban board with different sizes according to their CPCI index which makes it simple to understand the risk associated for individual capability on the backlog and unfinished work. Totals will tell how much work has gone or is going through the system (organisation) compared to the backlog.

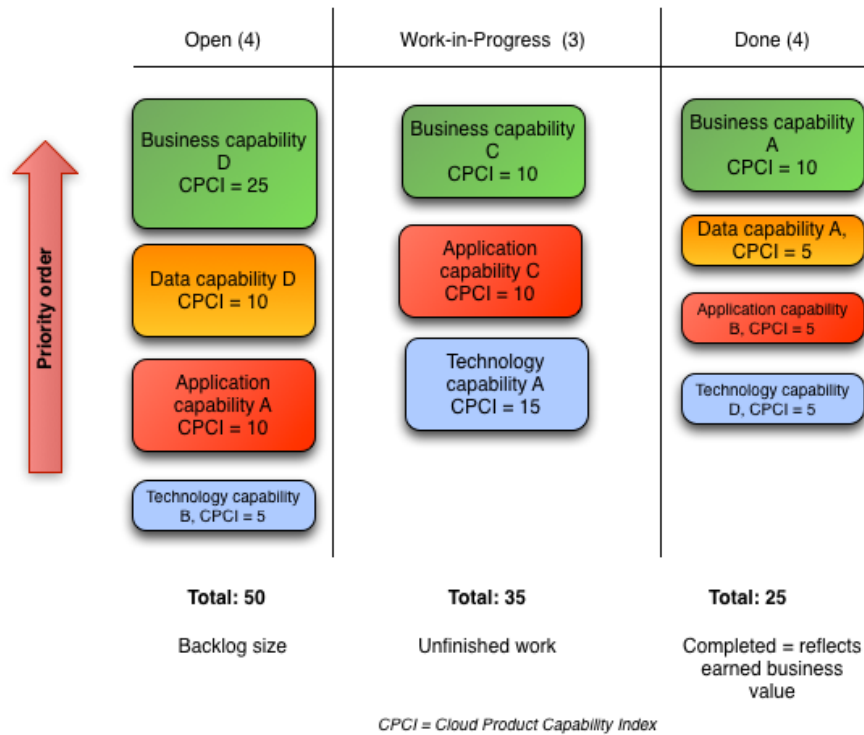


FIGURE 15 The development Kanban board

Important part of the study phase is to analyse together with the product team aforementioned aspects of the development progress. In addition to that the team should agree if individual capabilities in WIP status have achieved their definition of done and can be closed. Finally team should discuss if there is any organisational or any other changes which can impact any existing capabilities or capabilities under development. This is the continuous gap analysis step in the study phase. These findings should be recorded as status update of the product development and can also be communicated to the stakeholders.

The next and the final phase of the continuous development process is act phase. After this phase a new checkpoint for the product development is created. In this phase the team agrees the next step with three possible outputs: continue as planned, re-design or go-to market. Continue as planned means the team agrees there hasn't been any major deviations in the product development compared to the plan and continues to

the next plan phase where the capabilities are prioritised and capabilities missing CPCI and other information are updated.

If the team comes to the conclusion that the product and the plan needs re-design then this triggers a new planning phase where product vision is updated and needed capabilities are re-evaluated based on the information found in the Study phase. It might be that some capabilities are scoped out or new limitations are added and capabilities are re-evaluated in order to maintain the deadline to the market. This defines a new target for the to-be situation.

The final option of act phase is the go-to market which is illustrated in the figure 16. At this stage all the needed capabilities are developed and continuous development process execution ends. This is the stage when product is ready to the go-to market and the situation is communicated to the stakeholders.

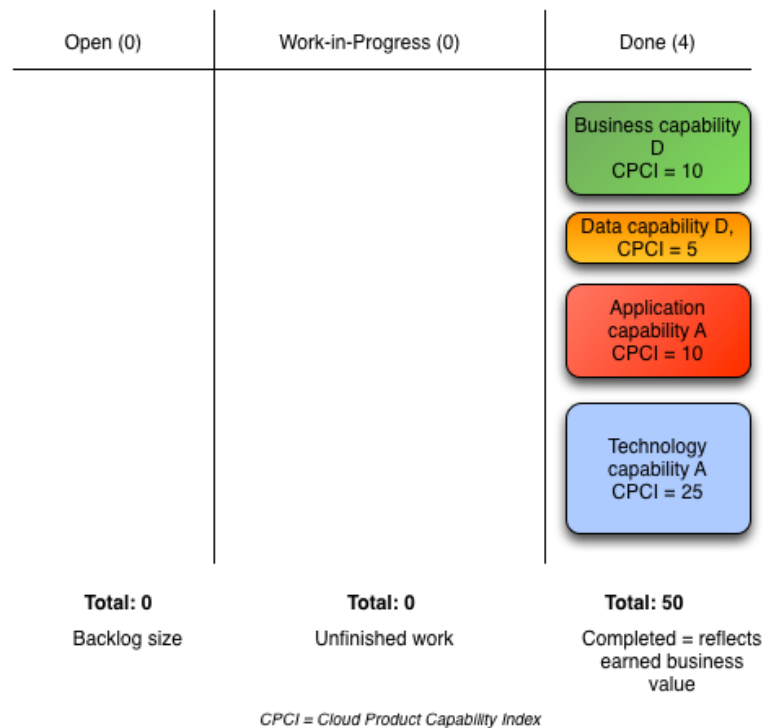


FIGURE 16 The continuous development process in the go-to market

4.1.4 Ontological analysis of EA & ITSM methods

Ontological analysis target is to build a default template and content for the cloud product capability assessment method. This content is used to build default capabilities for the method together with other sources and empirical findings which are described in the next chapter. Analysis is done by reviewing selected IT service management and enterprise architecture frameworks. Selected frameworks are ITIL v3 and COBIT 4.1 for IT service management and TOGAF 9.1 and Zachman v3 for enterprise architecture. These frameworks are found to be the most common and well-known frameworks in the IT management domain. ITIL is claimed to be even the *de facto* standard in the IT service management field in studies like Valiente et al. (2011) and Braun et al. (2007).

Analysis method was selected by reviewing previous researches and what different kind of research methods were used. Ontological analysis is theoretical study about existing frameworks and they should provide solid academic background for the template. In many studies (Järvinen et al., 2011; Valiente et al. 2011; Meertens et al., 2012) was referred to Wand and Weber (1993) original study: “On the ontological expressiveness of information systems analysis and design grammar”. This study was not any more available but Wand (1996) and Wand, Storey and Weber (2000) newer studies were available which define ontological research method. The ontological research method is used in the all aforementioned studies. Järvinen et al. (2011) explain origins of Wang and Weber studies which all are based on Bunge (1967) original study of ontologies.

Bunge (1967) research method wasn't used for this analysis because Wand et al. (2000) study was more widely used in IT domain (Valiente et al. 2011; Meertens et al. 2012). Wand et al. (2000) explains that their research method is based on ontology which is a branch of philosophy and it's meant for analysing constructs of reality. It seemed to be suitable method for analysing conceptual models like different IT management frameworks which try to define IT domain, assets and capabilities. Research is limited to review frameworks itself but not how well they define the reality.

Wand et al. (2000) have defined two key rules for the ontology analysis: construct overload and construct redundancy. Figure 17 illustrates these two ontology analysis rules. According to Wand et al. (2000) target is to reduce semantic ambiguity in ontologies and models by analysing their relations. Construct overload states there is two or more modelling constructs for one particular ontological construct. Correspondingly in the construct redundancy there is one construct which can be mapped to multiple other ontological constructs. (Wand et al, 2000). This method is utilised to analyse different frameworks on ontological level.

Analysis process is split to three phases. The first phase is ontological analysis between selected IT service management frameworks. The second phase is analysis between selected enterprise architecture frameworks. The third phase is analysis between IT service management and enterprise architecture frameworks capabilities and definition if capability will be selected into the cloud product assessment. There is also additional definition added how the capability should be seen from the product viewpoint.

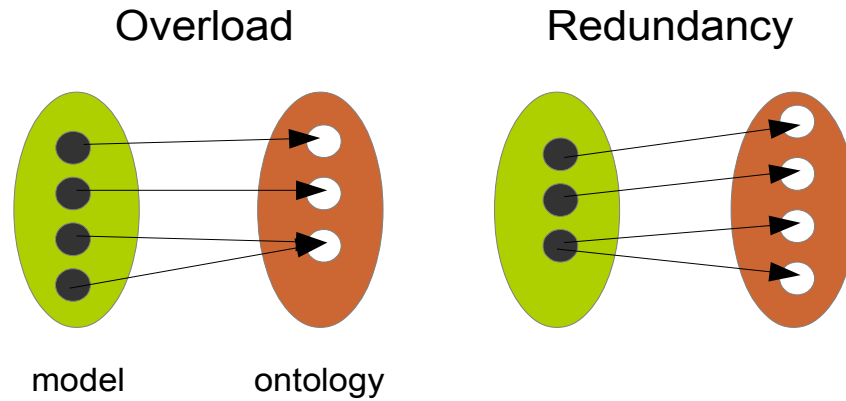


FIGURE 17 Ontology and model analysing rules according to Meertens et al. (2012)

IT Service Management

IT Service Management frameworks are process oriented frameworks and processes are divided to different process domains. ITIL has four different process domains and Cobit has five different process domains. Frameworks have different viewpoints and targets from the service management point of view as explained in the chapter 3. Therefore ontological analysis is not done between different process domains. That would need deeper analysis of different process domains, their ontology and relations between different frameworks. This is out of this thesis scope.

At ontological level processes have same kind of hierarchy on both frameworks: domain \rightarrow processes. Therefore ontological analysis is done at process level between frameworks. Cobit is considered to be higher abstraction level framework and therefore ITIL processes (model) are analysed against Cobit processes (ontology). Results are shown in the attachment 1: IT Service Management frameworks ontology analysis results.

Results show that ITIL is missing 21 of Cobit processes on ontological level. Correspondingly Cobit is missing 10 of ITIL processes. Frameworks have 13 processes in common. Interestingly there is only one Cobit process which overloads multiple ITIL processes.

Enterprise architecture

Enterprise architecture frameworks are evaluated in the same way than IT Service Management frameworks. TOGAF is considered to be more comprehensive than Zachman framework and therefore Zachman views (model) are evaluated against TOGAF architecture domains (ontology). Results are shown in the table 2.

It can be seen from the results that basically Zachman views map pretty well with TOGAF domains which is not a big surprise. Only exception is the data architecture which seems to be missing from Zachman framework. Frameworks could be analysed also other way around and data architecture could be marked to be redundant with all Zachman views because in the Zachman framework different views define different artefacts which should be documented. Analysis was relying on TOGAF description of data where the data is pervasive or transformational for the whole enterprise which seems to be missing from the Zachman framework.

TABLE 2 Enterprise architecture frameworks ontological analysis

TOGAF domains	Zachman views	Ontology problem
Business architecture	Scope contexts Business concepts	Overload
Data architecture		Missing
Application architecture	System logic Technology physics	Overload
Technology architecture	Tool components Operations instances	Overload

Ontological relations of EA and ITSM frameworks

Ontological relations between EA and ITSM frameworks are shown in the attachment 2. The basis of the ontological analysis is TOGAF architecture domain definitions (TOGAF, 2011). IT service management framework processes which don't map with TOGAF domains were omitted.

Vast majority of both IT service management framework processes map well with TOGAF business architecture. This outcome is not big surprise because both frameworks claim to be very well business aligned. Interestingly both IT service management frameworks seems to be lacking data architecture aspects. Both have only one process to address this architecture field. Analysis is done only on high level so basically most of data architecture aspects are implicitly handled on process level in their results and outputs.

IT service management frameworks seems to handle and address TOGAF application architecture domain pretty well and processes are aligned at ontological level. This is not a big surprise either because IT service management frameworks are built for managing internal IT operations which are generally concerned to be application operations.

4.1.5 Assessment template for the assessment tool

Assessment template define the key capabilities for the cloud product capability assessment method and the default content for the assessment tool which is defined in the chapter 4.2.2. The assessment template capabilities are defined from the product viewpoint as explained in the chapter 4.1.3, not for example internal process structure point of view. Target with the assessment template is to guide the cloud product managers to

define and recognise needed capabilities for a new cloud product. Assessment template and default capabilities are listed in the attachment 3.

The main source for the assessment template capabilities is the ontological analysis of EA and ITSM frameworks as explained in the previous chapter. Capabilities which are selected to the template get their default names from ITIL because it's considered to be the most well-known IT framework. Only the most common capabilities were taken into the template. All mostly internal process capabilities like strategy or portfolio management were scoped out because they are not considered to be relevant for the product development, seen by the customer and these aspects are generally addressed in other domains of the product management.

Ontological analysis provides a good source for the cloud product manager to consider if more capabilities are taken into consideration during the product development like security incident management or service continuity management. These are common among many cloud products, especially enterprise products but are not taken into the default template. If these requirements are important for customers then they should be taken into scope of the product development, especially after going to the market and acquiring customer feedback. Nothing prevents them to be taken into the first product version scope.

Each capability has also a description which explains why these selected capabilities are important from customer and the product point of view. Description should guide the product manager and the product team to evaluate if such capability is needed and should be taken into the scope.

The second source for predefined capabilities are literature review done in the previous chapters. For example business capabilities include different operational models which are important from business agility as well as IT and business alignment point of view, for more information please see chapters 2.5 and 3.

Third source for the default capabilities are researcher experience on building the cloud products. These capabilities are also seen in the literature review like marketing or sales material but those are highlighted in the template in order to guide the cloud product manager to think if the capabilities are needed. Some capabilities like the tenant provisioning can be understood from the cloud business and architecture definitions but are not always clear for the cloud product managers. If any of the predefined capabilities are not needed they should be removed from the assessment but they are considered to represent common capabilities across multitude of different cloud products.

4.2 Method development

Method development explains how the cloud product assessment method is developed based on the method design explained in the previous chapter. The output of the method development is the artefacts needed to execute the method. The method has two kind of artefacts: group assessments and assessment tool. Group assessments explain the setting and structure of group work assignment according to the capability development framework. The assessment tool is the tool that should be used to document, visualise and manage the product life cycle from inception to go-to market.

Example case

Example case shows how the cloud product capability assessment method and tool could be used through the new product development from inception to go-to market on an ideal case. The example case tells about fictional company called *Live events*. Live events is a global event organiser. They have business and leisure events product lines and professional services unit which provide needed resources from event planning to event organising and marketing. Company organisational structure is shown in the figure 18.

Company has so far sold event services to direct customers. Company CEO James has noticed that customers need also online ticketing services which company can't yet provide. James would also like to increase revenue by utilising more effectively existing supplier network. Therefore he has hired a new product manager Lisa to build a new cloud product to achieve these strategic targets.

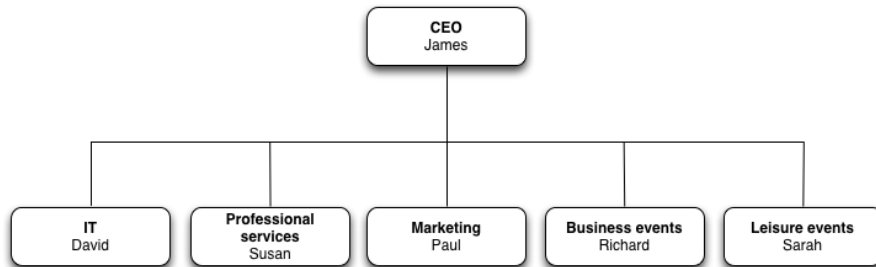


FIGURE 18 Live events organisation

4.2.1 Group assessments

The cloud product capability assessment has three different kind of group assessment activities explained in the design phase. Design activities include initial design of the product vision including the first capability analysis and updating existing design and altering capabilities accordingly. Planning activities include planning the needed capabilities. Evaluation activities include tasks related to evaluating the progress and changes in the organisation as well as deciding the next increment (act phase) direction.

According to the capability development framework and the continuous development process there is two phases where designing happens. Initiation phase is longer design activity compared to redesign. Inception phase combines different design activities like understanding AS-IS situation and defining the product vision with the business model canvas. Developing and framing the product idea to more concrete product concept is unknown and unpredictable route according to Osterwalder et al. (2010) and Geracie et al. (2013).

Design attitude is according to Osterwalder et al. (2010) management of design process where focus is finding outstanding design instead of making difficult decisions of alternative routes. This is well aligned with conceive phase of product life-cycle (Geracie et al, 2013) where product concept and supporting business plans are de-

veloped. Before aiming at designing the product the product manager should find people with different backgrounds and experience to join to the product team and get the commitment to develop the product idea (Osterwalder et al., 2010).

Because route through initiation phase can vary significantly between companies and product ideas only general guidelines are given for design workshops. Idea generation and building the understanding of the product most likely takes significant time so there should be enough calendar time reserved for designing and the product team should not rush into plan phase. Initiation phase can also include all kind of internal documentation and planning but it's out of the scope of this thesis. In this phase focus is building the product vision and defining the capabilities needed for the product before planning and development starts.

Based on this general guidelines of the design workshops are given in the table 2. Target with the design workshops is to build the knowledge of the product within the product team and update assessment tool with the details agreed in the workshops. There is general discussion topics given for the product manager to guide discussion towards the outcome. The product manager needs to tailor these for each session in order to incrementally develop the product concept. There is lot of guidance given how to keep effective workshops or the product manager can follow for example the design process defined by Osterwalder et al. (2010). Design workshop structure is applicable also for redesign activity which might be needed during the continuous development process. If this is the case redesign impacts most likely only some part of the product vision and therefore it can be combined with other planning activities.

TABLE 3 Design workshop

Workshop discussion targets	Workshop outcome
<ol style="list-style-type: none"> 1. Understand AS-IS situation <ol style="list-style-type: none"> 1. What is our business status today 2. What are our business pain points 3. What are competitors doing 4. What are market trends 2. Define TO-BE situation <ol style="list-style-type: none"> 1. Where we want to go 2. Why we want to go there now 3. What if we don't go there 3. Analyse gap between AS-IS and TO-BE <ol style="list-style-type: none"> 1. What are new capabilities product vision needs 2. How we should extend existing capabilities to support product vision 3. Which existing capabilities support product vision and need only minor changes (replacement investment) 	<ul style="list-style-type: none"> • Updated product business model into the assessment tool • Updated list of capabilities with definitions and limitations into the assessment tool • Record discussion outcomes into the assessment tool • Save sketches or detailed design documentation if done during the workshop to other knowledge management system for future use

Planning workshop completes planning activities related to capabilities evaluation and prioritisation for implementation. In the first workshop the base capability must be selected and other capabilities are relatively evaluated against to that. It's recommended to

keep planning workshop time boxed because planning poker needs intensive discussion and focus from the product team.

Planning workshop execution steps and outcome is explained in the table 4 with proposed 1,5 hour time box limit. With the proposed time limit roughly ten capabilities could be evaluated during one workshop. It's a good option to start with application or technology capabilities and then proceed with other capabilities in the later workshops. This also gives better oversight for the product team how product will be working from the customer point of view and what kind of problems it's built to solve. Workshops can also be organised around one domain like data or business capabilities.

The key part of planning workshop is playing the planning poker. The planning poker is organised around an individual capability. The planning poker starts with the product manager explaining what is expected from the capability from customer point of view. Then team asks questions from the product manager related to the capability. Target is to restrict, clarify and understand collectively what is expected and what can be scoped out. This first phase is time boxed to three minutes because it's not possible to know all the details at this stage. (Cohn, 2005.).

In the second phase the product team evaluates each capability relatively to the base capability. Target is to evaluate overall size (scope) of the capability not exact amount of work or complexity. Team members do that by everyone selecting a number from planning poker cards that represents the size of the capability but the card is not yet shown to others. When everybody has selected their size estimate then the cards are turned upside down on the table at the same time. This way anchoring can be avoided. Anchoring is a cognitive bias when one of the participants gives evaluation then others are "anchored" to this and can't do objective analysis. (Cohn, 2005.).

When cards are turned available the distribution of values are reviewed. Product team member must select size from the cards or if the size can't be evaluated then she will select a card with ? sign which represents that there is too much ambiguity that size couldn't be evaluated. If everybody selects the same value for the capability then it's selected for the size. (Cohn, 2005).

Typically there is different values selected and therefore team members who selected the biggest and the smallest value explain their reasoning and open questions behind the value. Then the product manager answers to the questions and concerns. In this phase it's important to remind that size is evaluated relatively to the base capability if there is lot of ambiguity in evaluations within the team or many question mark cards on the table. Now the team must decide if they select one of values on the table as size for the capability or if they want to replay the poker and after that select the size. (Cohn, 2005.).

TABLE 4 Planning workshop

Workshop execution	Workshop outcome
<ul style="list-style-type: none"> • Evaluate capabilities (1 hour time box) <ul style="list-style-type: none"> ◦ Product manager explains a capability and team asks questions (3 minutes) ◦ Play planning poker for the capability <ul style="list-style-type: none"> ▪ Persons giving the lowest and highest estimate explain their reasoning (2 minutes) ▪ Update restrictions, limitations and assumptions for the capability ▪ Replay if needed ▪ Agree together size (scope) for the capability ▪ Start evaluation with the next capability ◦ Update capability description (acceptance criteria), investment type, status and definition of done • Review and discuss the cloud product assessment status (15 minutes time box) • Prioritise capabilities for implementation (15 minutes time box) 	<ul style="list-style-type: none"> • Updated list of capabilities with all the details • Updated priority order for the capabilities • Select base capability (only in the first workshop) • Select development cadence (only in the first workshop)

Facilitation and time boxing are important parts of the planning poker. Otherwise team can get stuck to discuss about details of one individual capability. The base capability is selected on the first planning workshop either discussing about one simple or relatively complex capability and assigning a size for it. The facilitator shouldn't participate on the planning poker in order her to influence to the evaluation process. Typically facilitator can be the product or project manager who should get others committed to the delivery of the capabilities. (Cohn, 2005.).

When the capability is evaluated and size selected team should fulfil rest of the capability details. First they need to write down capability definition (acceptance criteria) and assumptions, limitations and other details as description. Then team needs to decide investment type and based on these details CPCI is calculated for the capability. Also capability status and definition of done should be updated at this stage. Definition of done explains when capability is agreed to be ready. There can be general definition of done for one domain e.g. application capabilities or then it's tailored for each capability. At this stage also owner can be defined for the capability.

At the end of the planning workshop there is two important tasks for the product team. First the team should focus on reviewing assessment status: how many capabilities are evaluated, how many capabilities needs still evaluation, whether all the findings are documented and so on. Target is to build consensus and overall understanding of the planning progress.

The final part of the planning workshop is prioritisation of the capabilities. This should happen around the Kanban board so that all capabilities can be seen. Then the team should discuss which are the next capabilities which should be implemented. This discussion should be guided towards available capacity of each capability owner and whether they can start implementation of a new capabilities. At this stage it should also be clear if there is dependencies between the capabilities, like marketing material can't be done before all the needed application capabilities are implemented.

The Kanban board should make it clear if there is already many capabilities under implementation and whether it's reasonable to start implementation of new capabilities. Based on this analysis relative priority is given to the open capabilities. Outcome of this is prioritised list of the product capabilities or prioritised backlog like the term is used in agile software development methods.

There is no planned group assessment in the *do* phase of the continuous development process. In the *study* phase the product team is tasked to analyse progress of the development. Like Anderson (2012) explains understanding the individual capabilities progress on the Kanban board and work in progress are the important characteristics of the (Kanban) system. The team should focus on optimising the throughput for the product and all the capabilities.

In the table 5 is explained *study* assessment execution guidelines and outcome. Effectively the *study* phase outcome is audit trail of changes for the capabilities and recognised changes in the organisation which will impact the product development.

TABLE 5 Study execution

Study execution	Study outcome
<ul style="list-style-type: none"> • Analyse progress <ul style="list-style-type: none"> ◦ How well targets and definition of done for each capability is achieved ◦ Is there any impediments for progress ◦ Can individual capabilities be closed • Organisational changes <ul style="list-style-type: none"> ◦ Is there are any changes that impact capabilities the product needs 	<ul style="list-style-type: none"> • Records of progress • List of impediments • List closed capabilities • Changes in the organisation and impact to capabilities

Final task in the continuous development process is the act phase. In this phase outcome of study phase is checked and conscious decision is made whether process continues as planned, does the product need redesign or is the product ready to go-to market. It's important to record this decision with the study phase outcome to have disciplined way of working. This will also finalise the continuous development process increment and create a new checkpoint for the product development. In the table 6 is shown an example way to record different phases outcomes and a checkpoint (increment #2) of a product is highlighted.

TABLE 6 Continuous development process checkpoints

Increment	Design outcome	Planning outcome	Study outcome	Act decision
#1				
#2				
Go-to market				

In order the product manager or project manager effectively communicate the cloud product capability assessment method including the capability development framework, the continuous development process and different group assessments these instructions are turned into the Cloud product assessment guidebook. The guidebook should be used to educate the method and to showcase process steps to the product team. For more information see the Cloud product assessment toolkit in the attachment 4.

4.2.2 Building the assessment tool

Information systems consist of people, software and data which support operations, management and decision making. The assessment tool is the software component and data repository for the cloud product capability assessment method. Software engineering is an academic study of software development and maintenance and it's highly relevant for the cloud product development because the cloud products consists of software components. In the assessment tool development focus is not going through all the aspects of the software engineering and how the assessment tool is built but focus is on how to use the assessment tool.

The assessment tool is built to support the cloud product capability assessment method. Target with the tool is to direct the method execution, document outcome of different phases and help the product manager to communicate product design and development to the product team and stakeholders. The assessment tool should answer research targets how to manage capabilities through product life cycle from inception to go-to market and especially how to show the progress during the product development.

In the previous chapters the cloud product assessment method design and group assessments were explained in details. These are the requirements for the assessment tool. The tool should fulfil these requirements and any limitations to fulfil those are explained. How the tool should be used as part of the assessment is explained through the example case company *Live events*. The tool complements the cloud product capability assessment method and is the final artefact needed before the method can be demonstrated in real life context. This should also finalise information system technical implementation for the method when there is method and tools ready for demonstration.

The cloud product capability assessment method is targeted to the product and project managers as well as to the product team. Therefore the tool must support the whole product life cycle management (PLM) process. The tool is intended to support the management of the cloud product domains (business, data, application and techno-

logy) and should not be mixed with domain specific tools like application life cycle management(ALM) tools. Target is to do governance for all of the domains in order to manage the product development holistically.

There is couple of non-functional requirements for the assessment tool. It should be easy to use, simple to adapt to different product development and organisation contexts and should support agile product development. Target is also to avoid any unnecessary overhead with domain specific tools like business process modelling tools. The last non-functional requirement is that the tool and technologies should be available as open source in order to support wider usage and adoption of the tool. This is also aligned with the research targets when the assessment method will be made available as open source with all the artefacts developed during the research process.

The assessment tool development started from analysing the requirements from the design of the method. Based on the requirement analysis and non-functional requirements for the tool two possible technology options were found. The first and the most obvious technology option was to build the tool by utilising cloud based services and open source programming languages like Java or Ruby on Rails. The second option was to build the tool with open source productivity tools like OpenOffice.

The first option had significant risks related to the researcher competencies and limited nature of resources in the thesis. Therefore small proof of concept (PoC) is built based on the requirements and the existing competencies. Building proof of concepts is common also in the agile software development methods in order to limit and understand technical risks related to new technologies or business domains.

There is wide variety of different open source software development tools and languages available. The nature of the thesis and researcher competencies narrowed technology selection of PoC to Play framework (<https://www.playframework.com>). The Play framework is well-known web application development platform built with Scala and Java and it supports both languages. In addition to that the framework is aimed to optimise developer productivity, support rapid application development and has wide variety of web application development templates available.

In couple of days simple PoC concept was built with the Play framework where end user could add needful information for capabilities. At this stage development focus shifted to analyse different visualisation options. There is lot of different Javascript based visualisation libraries available like D3.js, Flot or Chart.js. Different options were quickly evaluated but soon it is realised that integrating all these components together and making the tool available for method demonstration consumes significant time compared to the resources available and this ended the PoC development. Significant resource consumption is also related to the lack of competencies.

The second technology option was to utilise productivity tools which are well-known among knowledge workers. Based on the PoC and the research objectives to implement complete DSRM process productivity tools and namely OpenOffice Calc was selected as technology to build the tool. OpenOffice Calc is the spreadsheet application in the OpenOffice productivity suite. Otherwise there wouldn't be enough resources to execute demonstration of the method. The thesis target is not to analyse from the software engineering point of view different technologies and their feasibility for the requirements implementation but implement the cloud product capability assessment method from design science point of view.

Based on the design requirements and group assessments defined for the cloud product capability assessment method there was three views built into the assessment tool. *Capability assessment* view (tab in the Calc) shows product business model together with all the needed capabilities. This view was designed to support especially *initiation* and *plan* phases of the cloud product development framework.

In the capability assessment view the first section is the business model canvas (Business Model Canvas, 2015) which should be used to store product business model for latter use. This section is linked to the Business model canvas on the internet in order to facilitator to print the canvas for design workshops as explained to be the best practise in Business model generation (Osterwalder et al., 2010). Outcome of the design workshops should be recorded in the capability assessment like explained in the chapter 4.2.1.

The second section in the capability assessment view is the actual list of the cloud product capabilities and their properties. Properties include capability id, capability (name), description, investment type, scope, CPCI and status. These properties are seen in the figure 21 or attachment 5. Target with the capability assessment view is also to guide user to follow the cloud product capability assessment method. Therefore user can edit only the sections which need input from the end user. All other columns are locked in order to direct the end user to fill only needed parts and end user can't break for example algorithms.

OpenOffice Calc, like other productivity tools, have also couple of other important capabilities which makes it easier to build into the tool assistance how to use the method and guide how the method works. Each field in the tool which needs the end user input or is relevant for the end user has own guidance text which is shown when the field is activated even though the end user couldn't edit the information in the field. The guidance text explains for the end user why the field exists, what is the purpose of the field and how it should be used in the context of the cloud product capability assessment method. An example of guidance text can be seen in the figure 19.

The purpose of the guidance texts is to bring the cloud product capability assessment guidebook and the method design and instructions to the tool itself. This should support usability and adaptability when the end user doesn't need to switch between the instructions and the tool but the guidance texts help the end user to execute the process only with the tool. The other purpose of the guidance texts is to educate the academic background of the cloud product capability assessment method and how the field or it's content should be used during the product life cycle, like shown in the example below from the capability field in the tool.

“Short name for the capability which is shown and used to differentiate capabilities from each others and make it easier to recognise the capability in discussion. Capability name and the id of the capability should be used consistently also in other systems, like application life cycle management, to map capabilities development or maintenance to the product.

Default capabilities use common terminology from frameworks like IT service management which are basis for the Cloud product capability assessment method. This should make it easier for the product team to recognise how needed capability maps with existing capabilities in the organisation. All the default names can be changed.”

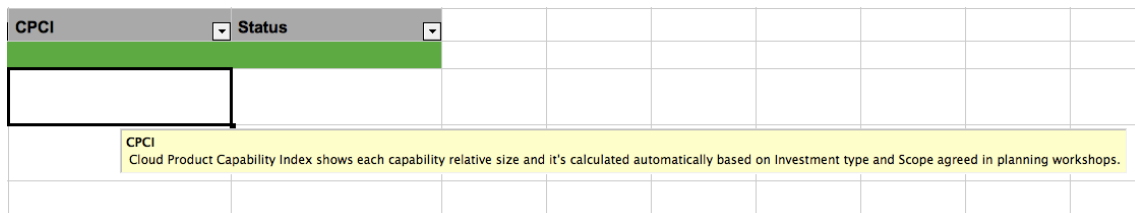


FIGURE 19 Assessment tool end user instructions

Other beneficial capability in the Calc is to build automated filters. These are used to give the end user possibility to filter capabilities with all the criteria like CPCI, scope or status. These filters should be used to narrow the capabilities shown and guide different discussions as part of the planning workshops. Filters help also other phases of the continuous development process like *study* when it makes possible to show only relevant capabilities.

Visualisation and visual management of the progress is one of the cloud product capability assessment method targets. Therefore in the bottom of capability assessment view it is listed totals of different capability properties. It is easy to see from the totals how many capabilities are in the product and how well the assessment is completed. In order to make assessment status even clearer the progress is visualised in one graph.

After creating the capability assessment view all the default capabilities from the assessment template are added. See chapter 4.1.5 for additional details. After adding the template as default capabilities into the tool there is already 27 different capabilities listed.

Testing of the capability assessment view was done with example company case by editing and adding more capabilities. Testing reveals that adding more capabilities and keeping everything in sync is not so simple. There is three options found how to proceed with the tool development. The option 1 is to build complex guidelines how to keep all the views, graphs and calculations in sync. The option 2 is to build complex macros which would do the syncing for the end user. The option 3 is to have hard coded limits in the tool and that way keep everything in sync.

The option 3 is selected in order to simplify the tool development and maintenance. The option 1 wouldn't fulfil non-functional requirements for the tool and option 2 is hard to achieve, maintain and it's also hard to export for other productivity tools because macro languages differ significantly and they might not support same functionalities. Based on the number of default capabilities in the template hard coded limit for each capability domain is set to 20 capabilities. This means that there can be added roughly 10 more capabilities in addition to the default capabilities and this limit is acceptable for the method demonstration. The hard coded limit can be relatively easily increased in the future but it needs understanding how the tool works.

It's also worth noticing that at this stage it becomes clear that productivity tools have technology limitations to build all the needed capabilities. These limitations weren't known when technology selection was done. The limitations aren't considered to be that significant that they would prevent the future development of the tool.

How to use the capability assessment view

Capability assessment view is supposed to be used through the whole life cycle of the product. Different use cases will be elaborated through the example case company *Live events* and how they could use the cloud product capability assessment method and how the assessment tool will assist them through the journey.

After joining to the Live events company Lisa (product manager) was tasked by James (CEO) to build a new cloud product to support existing business with new channels and also build new business by online ticketing service. Lisa has no former experience of event management but extensive experience on building cloud products. Therefore she starts interviewing all key stakeholders from each organisation unit. That way she builds knowledge of existing business and capabilities in the organisation.

During the interviews Lisa explains shortly her task and that the idea of the new product is on it's infancy and everyone can contribute by bringing new ideas on the table. Target is to build new revenue sources for all units. That way she knows she will get more open feedback and commitment from different leaders. After some time and interview rounds Lisa has basic knowledge of the organisation and it's capabilities.

At that time Lisa will start organising more formal design workshops according the cloud product capability assessment method. She'll first get named resources from each unit lead to build the product team around the new product. Because this is strategic initiative and includes new capability development leaders from IT and Marketing units join to the team. Other units name their development managers to be part of the product team. Based on design workshops the product business model is defined and it can be seen in the attachment 5.

Now the product initiation phase is concluded and overall situation can be seen in the capability assessment view summary. The new product is simply called as Live events in order to build a new online presence for the company. Summary of initiation phase can be seen in the figure 20.

After the initial workshops and before continuous development process starts the cloud product status can be analysed with the capability assessment view. In the example case (see the figure 20) there is altogether 33 capabilities which all has name and definition. During the initial workshop 14 capabilities out of 33 (42%) has investment type selected and even some capabilities has status added. By utilising the filters it's easy to see which capabilities are new, replacement or expansion to the existing capabilities. In the example case only 2 out of 14 capabilities has expansion type and rest of the evaluated capabilities are new for the company.

The assessment status view shouldn't be used at this stage to withdraw too many conclusions of the cloud product and it's development status or risks because none of the capabilities are evaluated in the planning workshop. This should only give overview of the initial design and if the product concept is ready to go to the continuous development process. Therefore it's important to showcase the product initial idea of business model and needed capabilities to the stakeholders. When stakeholders have committed to the product initial idea it's time to move to the continuous development process.

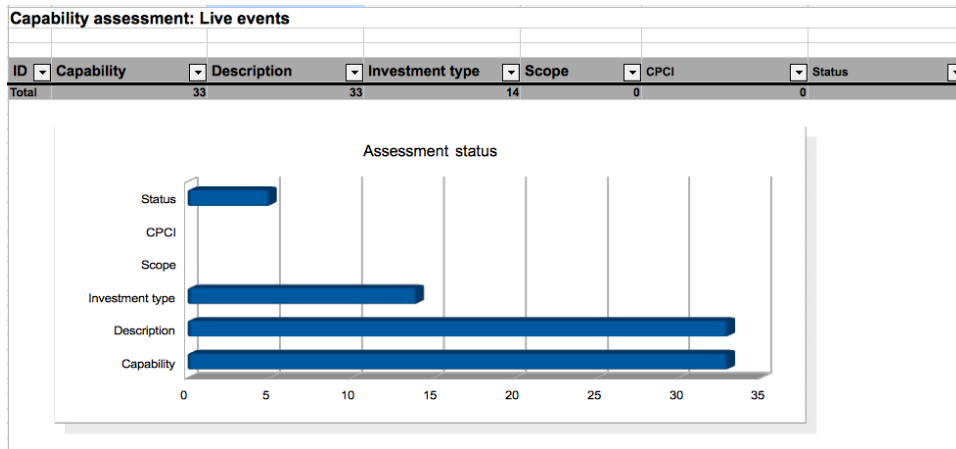


FIGURE 20 Live events status after initiation phase

The assessment tool has second view called *Development view* which is used to follow the progress of the development. The development view is intended to be used for all phases of the continuous development process. It is used to visualise individual capabilities size based on CPCI on the Kanban board as well as overall progress of the product development.

The development view shows following properties of all the capabilities: id, capability, definition of done, priority, owner, status and size. Id, capability, status and CPCI are automatically fetched from the Capability assessment view but other properties are edited on the development view. Definition of done is used to record when capability is considered to be ready e.g. when an application capability is acceptance tested or organisation built, resources signed and competence transferred for a business capability like incident management. The product team must agree on these definitions during the plan phase.

The development view has also important role in prioritising capabilities. Capabilities are simply prioritised by giving them rank which is number between 1 - 80. All the capabilities are then sorted from smallest to biggest based on the rank. The product team defines ranking whether it is linear from the most important (smallest number) to less important (biggest number). Other option is to use grouping on the rank so that all of the most important capabilities get the smallest rank e.g. 1, the second most important ones get the rank 2 and so on. In addition to the ranking each capability has assigned the owner who is responsible of the capability development according to the description and definition of done.

How to use the development view

The development view has same kind of filtering and summary capabilities than the capability assessment view. In addition to the these there is visualisation of individual capability and the progress of all of the capabilities. During the development view development there is another significant technology limitation found in the OpenOffice Calc. According to the design each CPCI index should show relative size of the capability. Visualising each possible sizes of a capability in the OpenOffice Calc isn't possible without complex macro development which was out-scoped already in the previous view development.

The Calc has possibilities to have three different styles based on the content which can be used in the visualisation of a capability. This limitation was used to build three classes for the capabilities according their CPCI index: small, medium and huge. Following mapping is used to map CPCI index to this new classification:

- Small: CPCI between 1- 15
- Medium: CPCI between 16 – 29
- Huge: CPCI over 29

The classification is done by calculating all the possible values of CPCIs based on the default risk parameters. After that CPCI values are split according the distribution in the way that 80% of values would go evenly to small and medium classes and rest to the huge class. This is not ideal but it provides a way to highlight that capabilities in the huge class are significantly bigger than the rest of capabilities.

After the initiation phase Lisa is ready to kick-off continuous development of Live events. She will follow the cloud product capability assessment method and calls up the first planning workshop for the product team. In the first workshop she will first explain the method they have already followed by utilising the guidebook for the cloud product capability assessment method.

*After shortly explaining the method she will share planning poker cards for the team and explains shortly the planning poker method. She emphasises that evaluation is relative and by any means it's not intended to evaluate actual work days needed. The team discuss shortly about the evaluation method and decides to select **Consumer registration** as base capability and assigns 3 as scope for the capability.*

At the end of the first planning workshop team reflects assessment and development status, see attachment 5 for details. The team notices there is altogether 9 capabilities which are evaluated and those capabilities are prioritised for development. Each evaluated capability owner is assigned and the first increment can start. The team agrees continuous development cadence to be 2 weeks.

After couple of increments the product development has progressed to the situation which can be seen in the figures 21 and 22. The figure 21 shows how powerful the development Kanban board is by showing development status in one view. There is only two items under development and couple of items are already finished. From the Kanban board can also be seen that different kind of capabilities can have significantly different definition of done criteria even though vast majority of them share the same definition.

Product: Live events

Development Kanban								
ID	Capability	Definition of done	Priority	Owner	Open (6)	WIP (2)	Done (4)	
T1	Supplier management	Suppliers evaluated and selected		1 David			6	
T5	Technical feasibility	Technical risks understood, competencies for development defined		1 David			16	
A1	B2B registration	Acceptance testing done		2 Lisa			26	
A5	Consumer registration	Acceptance testing done		2 Lisa			6	
A3	Release and deployment management	Acceptance testing done		2 Lisa		26		
B7	B2B incident management	Resources hired, trained and ready to serve first pilot customers		2 David		40		
A8	Event management	Acceptance testing done		3 Lisa	80			
A2	Backend integrations	Acceptance testing done		3 Lisa	12			
A4	Credit card integration	Acceptance testing done		5 Lisa	40			
B8	Consumer incident management	Resources hired, trained and ready to serve consumers in pilot country		5 David	40			
A6	User management	Acceptance testing done		6 Lisa	26			
A7	RFP management	Acceptance testing done		7 Lisa	19,5			
B1	Marketing model			Paul				

FIGURE 21 Live events development Kanban board

In addition to visual signalling on the Kanban board the development view has also the product development overall status. This overall view should be used jointly with the assessment status. The assessment status shows whole product backlog when the development status can be used to show only the development backlog and status. This separation is easy to achieve if individual capability development status is not updated to be *open* before it's wanted to be shown in the development view. Or then all capabilities can be setup with *open* status as soon as they are evaluated with the planning poker. In the latter scenario there is no major difference between product and development backlogs.

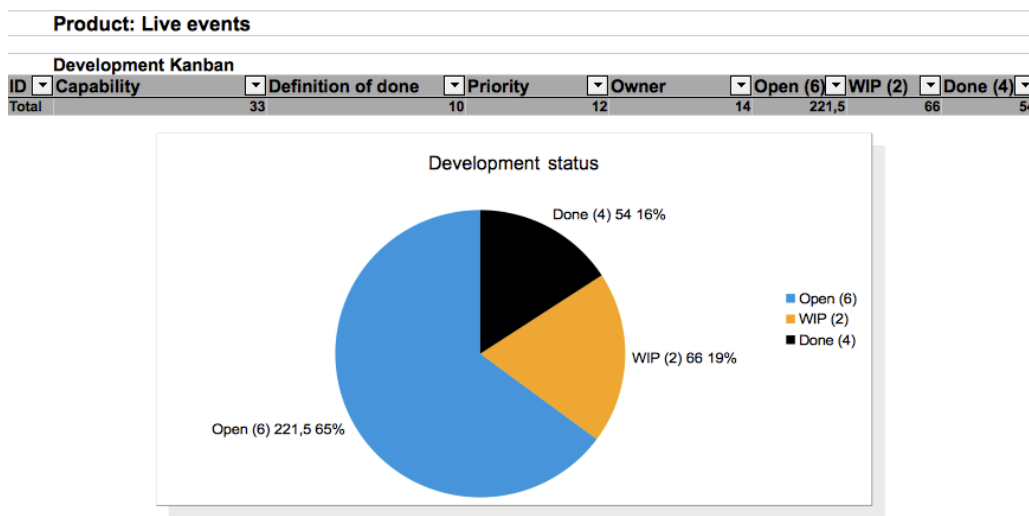


FIGURE 22 Live events product development status

How to use the checkpoint view

Product development status, direction and also product capabilities requirements can change over time as explained in the design of the cloud assessment method. It's hard to impossible record all the changes that can happen over time. The assessment tool has third view called checkpoints to record any major changes during the development. An example set of data in the checkpoint view can be seen in the figure 23.

Product: Live events

Continuous development change log				
Increment	Design workshop outcome	Planning workshop outcome	Study outcome	Act decision
1	33 capabilities from initiation phase	9 capabilities evaluated and prioritised	PoC (T5) proceeding well Credit card implementation can't start before backend integration	Continue as planned
2	NA	5 capabilities evaluated	PoC (t5) finished. Global rollout not possible.	Continue as planned
3	NA	8 capabilities evaluated, priorities updated	Finished: T1, A1, A5	Redesign go-to market tactic
4	No major changes to the product vision. Go-to market updated to consider only 2 pilot countries.	6 capabilities evaluated, B2B incident management (B7) prioritised to 2 group in order to hire and train support people	Reasonable amount of work in WIP. No major impediments.	Continue as planned
5				
6				
7				
8				
Go-to market				

FIGURE 23 Live events change log

The checkpoint view is meant only for highlighting major changes and decisions during the continuous development process. An example can be seen in the figure 23 in increment 3. Study phase has a finding recorded that global roll-out is not possible and therefore act phase has triggered additional design workshop. Nevertheless of the major decision made and change to the go-to market tactic there was no major change identified in design workshop for the next planning phase.

This kind of audit trail and change log is important part of project management methodology. Audit trail provides the product manager way to communicate afterwards the route the product development has taken and the changes on the way. It's also a sign of disciplined development and decision making process.

Final step in the continuous development process is to decide when product is ready for the go-to market. At that stage all the capabilities should be ready and agreed to be finished. Visualisation on all views should show that everything is completed and examples of this self-explanatory product development status is omitted.

4.3 Demonstration planning

In the previous chapters the cloud product capability assessment method design and artefact development is explained. According to Peffers, et al. (2007) demonstration of the design and solving one or more instances of the problem is important part of the DSRM process. Demonstration target is to evaluate and prove that the idea works. The design of the cloud product capability assessment method is done for the ideal case where the method would be taken into use in the early phase of a new product development. It's rarely the case, especially with new methods.

Assumption with the cloud product capability assessment demonstration is that the product development has already started or progressed some time. Also thesis

resources are limited and therefore there is no possibilities for formal evaluation of the method in multiple different context.

Method demonstration will be done with explanatory case study method. According to Järvinen et al. (2011) explanatory case study method tries to explain phenomenon from certain viewpoint. In this case the target is to explain if the method is understood and if after demonstration there is still interest to continue using the method in the real life context.

Method demonstration is split to three phases: The first phase is product manager interview to record the product vision. The second phase is execution of the method with the product team and the third phase is a simple survey for people who has participated either of the phases.

The first part of the method demonstration is the interview with the product manager, concept owner and or project manager. The interview target is to simulate actual initiation phase where product idea is originally formalised. The interview will be done with the assessment tool and at the same time collect product vision (business model) and the most important capabilities for the next phase of the method demonstration. Outcome of the interview should be the first revision of the cloud product capability assessment same way than it would be used in the initiation phase. At the same time researcher makes notes of the interview. The interview is time boxed to one hour in order to test adaptability and learnability of the method.

The second part of the method demonstration is a workshop with the product team. The workshop target is to simulate the first usage of the cloud product capability assessment method with broader audience (product team). The workshop is aligned with the actual *plan* workshop defined in the method. The workshop is time boxed to two hours in order to test learnability and adaptability of the method. The researcher is the facilitator of the workshop and makes notes afterwards. The workshop agenda is:

- Explain research method, academic background and workshop target (15 min)
- Review and discuss of the product vision (business model) and explain shortly identified capabilities from the interview (15 min)
- Evaluate capabilities (1 hour)
 - This phase should be explicitly the same than defined in the cloud product capability assesment method
- Review results (15)
 - In this phase researcher explains how the results should be analysed like it's explained in the method

The workshop can be executed multiple times simulating multiple checkpoints and planning sessions if suitable candidate is found. In that case research method explanation is removed and focus is on formally executing the plan phase. The final part of the method demonstration and data collection is a simple survey which is sent to the people participating to the method execution. Target with the survey is to complement data sources with more quantifiable data and acquire anonymous feedback of the method.

In the survey there is three different kind of questions: background information, structured questions and open questions. There are two scales used in structured questions. Likert scale is used for 8 out of 10 structured questions with the scale from 1 to 5 where 1 means that respondent strongly disagree with statement and 5 that respondent strongly agree with the statement. Likert scale is used in these questions be-

cause it's known have less cognitive burden and the survey questions from 4 to 11 measure more the benefits of the method which should be fact driven.

Semantic differences are known to be better for measuring affectivity especially in interaction research. The cloud product capability assessment method is heavily based on collaboration and interactions and therefore the survey questions from 12 to 13 utilise semantic differences in scales. These questions use also scale from 1 to 5 in order to calculate statistical figures for the results. Target with semantic differences is to find the overall feeling after the workshop and understand if people think the workshops are beneficial or not and would they like to use the same method in the future.

The final questions in the survey are open questions regarding what went well and what should be improved. There is also given a possibility to give any other open feedback. The survey questions are classified to six different classes. Survey questions from one to three are meant to collect background information of the respondents. Target is to classify participants role and main responsibility area.

The questions 4,5 and 7 are used to collect feedback how well the method and the workshop is understood and working from respondents point of view. The questions 6 and 8 are used to collect feedback of the collaboration and interaction within the product team. Visualisation is one of the core parts of the cloud product assessment method and the questions 9,10 and 11 are targeted to get feedback for different views in the tool and how the results should be interpret. The last two structured questions 12 and 13 are used to collect feedback of the method usefulness and willingness to participate to the future workshops.

The last three questions 14,15 and 16 are used to collect general open feedback from the respondents. All the survey questions, scale used and the detailed reasoning behind each of the survey questions are shown in the attachment 6: survey questions.

5 DEMONSTRATION RESULTS

The cloud product capability assessment method is demonstrated in a company called Basware. Basware is a global leader in financial software and transaction services. According to Basware strategy major new business growth is expected to come from SaaS model and development focus is on cloud based solutions. Recurring revenue target is 80% of net sales by the end of fiscal year 2018. (Basware's strategy for 2015 – 2018, 2015). This makes Basware as an ideal candidate for demonstrating the cloud product assessment method.

Researcher worked at Basware as a product manager during the thesis writing. In order to maintain research external validity research candidates are taken from the area where the researcher is not directly involved in the product management. The nature of a new product development is highly sensitive in any company. Therefore the actual results like business model or capabilities of the product can't be released.

The cloud product capability assessment method is demonstrated according the method demonstration guidelines described in the chapter 4.3 for one product which was under development during the thesis writing. Because there is no multiple cases studies available statistical analysis can't be executed for the results. The results are presented from explanatory case study perspective which try to describe the testing based on the researcher observations and actual results in different phases of the demonstration.

The first phase in the method demonstration is the product manager interview which is time boxed to one hour. In the interview researcher first explains the business model and how it's used to define the cloud product. Then the researcher and product manager fulfil the assessment tool and business model for the product. The business model isn't known by the interviewed product manager and almost the whole interview is spent on discussing around the product business model. At the end of the interview product business model and four application capabilities are defined for the capabilities assessment section in the assessment tool.

The second phase in the method demonstration is simulated *plan* workshop from the cloud product capability assessment method. The product manager is instructed to invite cross-functional team which would represent the product team for the workshop. There is no strict definition of what the product team would be or who should be invited but this is left to the product manager to decide.

The workshop is started with 15 people joining to the workshop. The people represent broadly different competency areas of the new product including but not limited to product management, R&D, support and delivery organisations. There is many more participants than expected which meant that the researcher must adapt the workshop structure a little bit in order that everybody could participate to the capability evaluation. It is decided on the fly that these 15 people will form pairs who will do the planning poker together.

The workshop is started according the agenda. The researcher presents shortly the method and academic background of the method in roughly 20 minutes. Then the product manager starts explaining the product vision with the assessment tool and previously fulfilled business model. The product vision discussion is ended up to lively conversation where almost all of the people participated. According the agenda product vision discussion should take only 15 minutes but it takes 45 minutes.

After the product vision and business model discussion the researcher shares planning poker cards to pairs and shortly explains the planning poker method to the participants. Some of the participants acknowledges the relative assessment method because it is earlier used in a R&D project. The change to the agenda is made in order to pairs to discuss first shortly (2 minutes) together about evaluation before the planning poker results are shown collectively.

The capability evaluation starts with selecting the base capability. It takes significant time to agree with the team which capability is the base capability and how it's scope is evaluated. Eventually team plays planning poker twice to define the scope of the base capability. After the base capability selection the planning poker goes mostly according the rules defined in the planning workshop structure. Only difference is that pairs first discuss about the scope and then it is shown to others at the same time. During the planning poker the researcher, who worked as facilitator, realises it's important to have strict time boxes for each phase. Otherwise discussion around one capability can go too much details or even off-topic and time passes by.

The planning poker phase is ended after 45 minutes because workshop time box is full. There is five capabilities evaluated which means that it takes on average almost ten minutes to evaluate a capability. The evaluated capabilities are related only to the application domain. There is no time to review the assessment results. The researcher sends the assessment results afterwards with email to the workshop participants. In the same email research explains how the assessment results should be interpret. The assessment results are interpret in the same way than it's explained in the chapter 4.2.2 how different views should be used.

The researcher's observations from the workshop:

- The group is too big and not fully representative of the product team
- Pair assessment works surprisingly well. No major problems for the planning poker
- The first evaluation (base capability) is the hardest one to do. After that evaluation and planning poker goes smoothly
- There is significant risk of confusion when the capabilities under implementation are taken to the evaluation. People aren't sure should they evaluate the scope left or total scope. This should be avoided.
- Pair evaluation needed to loosen the time boxes for capability evaluation in order to facilitate pair discussion before playing the poker
- No time to review and discuss about the results of the workshop

Survey results

The survey is sent to all 15 participants shortly after the workshop. The survey structure and questions are described in the chapter 4.3. There is originally one week response period but it is extended to two weeks in order to get more responses. Altogether 10 (67%) out of 15 participants responded to the survey. Survey respondents can be seen in the figure 24. Only one of the respondents participated to the both sessions (interview and workshop).

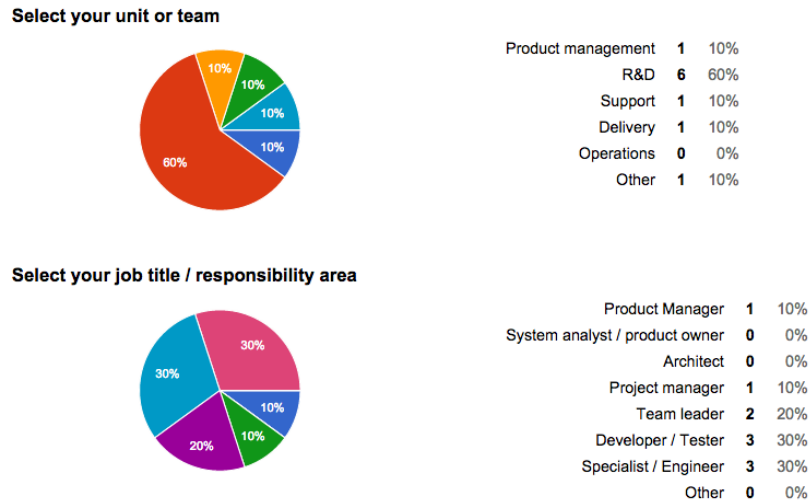


FIGURE 24 Survey respondents

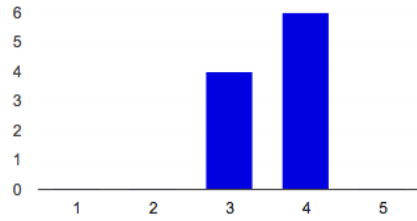
The survey responses are shown according their classification detailed in the demonstration chapter 4.3. In the figure 25 can be seen the responses related how well the different methods and the workshop are understood and working from respondents point of view. Visualisation is one of the core parts of the cloud product capability assessment method and responses to visualisation methods regarding the product development status are seen in the figure 26. Responses to the collaboration and interaction within the product team are seen in the figure 27. The last two questions are used to collect overall feedback of the cloud product capability assessment method usefulness and willingness to participate to the future workshops and the results can be seen in the figure 28.

In addition to the actual responses also the key statistical figures of the responses are shown in the table 6. There is no further statistical figures like Cronbach alfa or t-test results calculated because there is only one set of responses and only ten responses available. Open feedback is also analysed and classified in the end of the results section.

TABLE 7 Survey results statistical analysis, N = 10

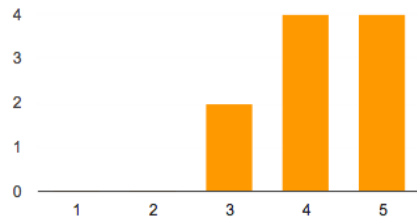
#	Question	Average	Median	Standard deviation
4	Workshop targets were clearly presented	3,6	4	0,516
5	Group assessment (planning poker with cards) method was easy to understand	4,2	4	0,789
6	Discussion around product business model helped me to understand product vision	3,8	4	0,632
7	After workshop I understand what a capability is and what kind of different capabilities are needed to bring product to market.	3,7	4	0,675
8	Scope analysis with cards and discussion with others helped me to understand expectations and development complexities	4	4	0,471
9	Assessment status was clearly presented	3,6	4	0,516
10	Product development status was clearly presented	3,5	3,5	0,527
11	Individual capability scope and status was clearly presented	3	3	0,816
12	For me the workshop and group assessment was... Scale from 1 to 5 where 1 represents useless and 5 useful	3,2	3	1,033
13	Future workshops would be... Scale from 1 to 5 where 1 represents waste of my time and 5 help to commit to product development	3,4	3	1,08

Workshop targets were clearly presented



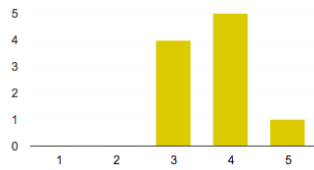
Strongly disagree: 1	0	0%
2	0	0%
3	4	40%
4	6	60%
Strongly agree: 5	0	0%

Group assessment (planning poker with cards) method was easy to understand



Strongly disagree: 1	0	0%
2	0	0%
3	2	20%
4	4	40%
Strongly agree: 5	4	40%

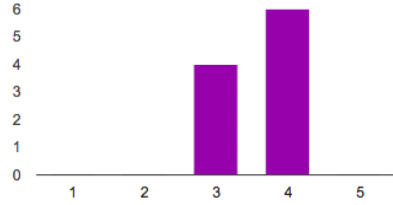
After workshop I understand what a capability is and what kind of different capabilities are needed to bring product to market.



Strongly disagree: 1	0	0%
2	0	0%
3	4	40%
4	5	50%
Strongly agree: 5	1	10%

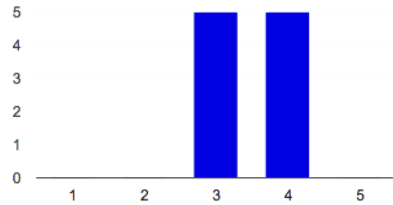
FIGURE 25 Survey responses regarding different methods used

Assessment status was clearly presented



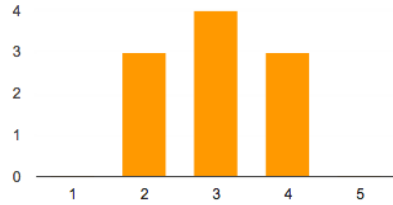
Strongly disagree: 1	0	0%
2	0	0%
3	4	40%
4	6	60%
Strongly agree: 5	0	0%

Product development status was clearly presented



Strongly disagree: 1	0	0%
2	0	0%
3	5	50%
4	5	50%
Strongly agree: 5	0	0%

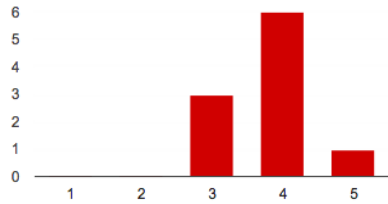
Individual capability scope and status was clearly presented



Strongly disagree: 1	0	0%
2	3	30%
3	4	40%
4	3	30%
Strongly agree: 5	0	0%

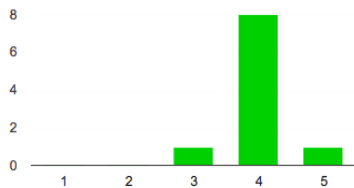
FIGURE 26 Survey responses for the different visualisation methods

Discussion around product business model helped me to understand product vision



Strongly disagree: 1	0	0%
2	0	0%
3	3	30%
4	6	60%
Strongly agree: 5	1	10%

Scope analysis with cards and discussion with others helped me to understand development expectations and complexities



Strongly disagree: 1	0	0%
2	0	0%
3	1	10%
4	8	80%
Strongly agree: 5	1	10%

FIGURE 27 Survey responses regarding collaboration and team level interactions

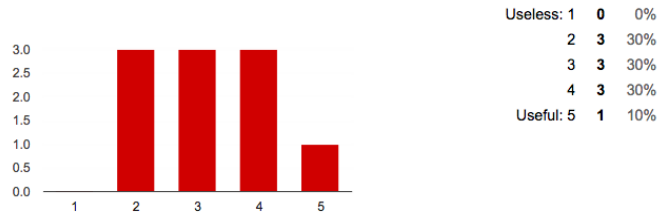
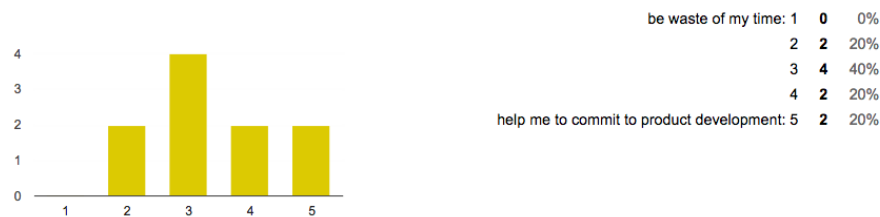
For me the workshop and group assesement was**Future workshops would**

FIGURE 28 Survey results regarding usefulness of the workshop

In the survey there was five responses given to the question what worked well and six responses to the question what should be improved and one response to other comments.

The main themes for the responses what worked well:

- Collaboration together: 4 responses
- Different viewpoints to the product: 2 responses
- Evaluation method (planning poker): 2 responses

The main themes for the responses what should be improved:

- Detailed specifications needed: 3 responses
- Time boxing didn't work: 2 responses
- Method shouldn't be used in the middle of development: 1

6 DISCUSSION AND CONCLUSIONS

The research had two targets set upfront. The first target was to build new knowledge for product management how IT service management (ITSM) and Enterprise architecture (EA) frameworks can be integrated and support the cloud product management. The second target was to build a new method called cloud product capability assessment method which should support the cloud product management and life cycle management. Focus in the method development was on how to build new products from inception to go-to market. The method was targeted for product and project managers as well as architects working with the cloud products.

There was two main research questions made for the research: what kind of capabilities a cloud product has and how to manage needed capabilities from inception to go-to market. Two main research methods were identified to answer the research questions. The first research question should be answered by doing a literature review of cloud products from business and IT viewpoints. The second research question should be answered by building the new method with Design Science Research Method (DSRM) process.

Literature review

The literature review was split to two viewpoints: business and IT, and how these could be brought closer to each other. First short instruction from academic research was given for a product and then the product definition was used to expand National Institute of Standards and Technology (NIST) definition of cloud products. The new definition for the cloud product can be considered to be broader and more holistic approach to define the cloud product in the context of business management. The definition takes also into account all product aspects when the NIST definition focuses mostly on the core benefits of the product.

Business viewpoint was expanded with short introductions of the cloud product management, business modelling and business capabilities. At the end of the business viewpoint explanation was given why software business and especially the cloud product business needs agility. The business viewpoint can be used as a general guidebook for all who are interested in what is a cloud product and cloud product management. Intended audience for this guidebook are the product managers as well as technology oriented people who want to know more about cloud product management.

IT viewpoint explains shortly what IT service management (ITSM) and enterprise architecture (EA) frameworks are and IT and business alignment problem. IT viewpoint is targeted to product and business managers to give quick overview and basic knowledge of the benefits and why especially the cloud product managers should know the IT management frameworks. This part of literature review is limited in details of the IT management frameworks and should be relevant only for people who doesn't know the IT management frameworks and IT viewpoint to the capabilities.

Researcher recommends to get more deeper knowledge of the IT management frameworks if they are critical for the product manager working with the cloud products. One good example could be a product manager responsible of cloud based IT service management solutions. Further analysis of the IT management frameworks was done in the chapter 4.1.4 where ontological analysis between ITSM and EA frameworks was done as part of the cloud product capability assessment method development.

Method development

The cloud product capability assessment method had ambiguous targets to build a new agile product development method for the cloud product management. During the design of the method a product viewpoint was developed from enterprise architecture principles to holistically manage all the capabilities the cloud product needs. This viewpoint takes all the product layers into account and should be relevant for everybody working in the cloud product and IT management.

The method design expanded enterprise architecture analysis method from analysing current status (AS-IS) and target status (TO-BE) to be more agile and flexible into changing situations in the organisation. Even though for example TOGAF explains the architecture development is ongoing process it's not always clear how the architecture is evolving and adapting to the changing context. This was made more explicit and easy to use in the capability development framework and the continuous development process.

Based on the design of the method artefacts a small and simple toolkit was built to manage and visualise capabilities during the product development. The toolkit development was limited in functionality because of selected technology had limitations and there was no time to build the tool with alternative technologies. This wasn't considered to be a major blocker for demonstrating the method.

The cloud product assessment method was demonstrated in one real life context and results are shown in the chapter 5. Major conclusions can't be made of the first demonstration round because it had limitations in the target audience. Based on the survey results after the workshops it can be seen that the agile way of working increased collaboration inside the team and brought even new insights to the team that has existed for a long time. Biggest variation in the survey responses is on the visualisation of the cloud product development status. This shows that the tool and the method doesn't yet provide good enough visualisation and visual management to the cloud product development.

Before broader conclusions of the method will be made it should be tested in another context. Therefore the method and developed tools are made publicly available as an open source tools, see attachment 5. From design science point of view this thesis concludes the first iteration of the cloud product assessment method development but the researcher has already many further development areas in mind which are elab-

orated in the next chapter. The researcher also hopes to get feedback and further development topics from the community by sharing the research artefacts and findings.

6.1 Further research topics

As explained already in the thesis introduction and research objectives the thesis scope was limited from the product life cycle point of view. The next future research topic should be how the cloud product assessment method is thoroughly expanded to cover the whole product life cycle. This might impact how the continuous development method is used and it also brings other complexities like life cycle management of different capabilities into consideration. The capability development framework and the continuous development process should be applicable to the product increments.

Another future research topic would be to build proper tools to utilise the cloud product assessment method in other context. The current version has limitations, especially on visualisation and needs too much knowledge of the technology before the tool can be taken into broader use.

REFERENCES

- Agile Service Development. Combining Adaptive Methods and Flexible Solutions*. Editor Lankhorst, M. 2012a. Berlin-Heidelberg: Springer-Verlag.
- Anderson, A.(2010). *Kanban: Successful evolutionary change for your technology business*. Sequim, Washington: Blue hole press.
- Anderson, D.(2012). *Lessons in agile management. On the road to Kanban*. Sequim Washington: Blue Hole Press.
- Basware's strategy for 2015 – 2018*. (2015). Picked up 1.4.2015 from address <http://www.basware.com/about-us/strategy>.
- Benlian, A., Koufaris, M., Hess, T.(2010). The role of saas service quality for continued SaaS use: empirical insights from SaaS using firms. *Thirty First International Conference on Information Systems*, St. Louis.
- Braun, C., Winter, R.(2007). Integration of IT Service Management into Enterprise Architecture. *ACM March 11-15*.
- Bunge, M. (1967). Scientific research II. The search for truth. *Berlin: Springer-Verlag*.
- Business model canvas*.(2015). Picked-up 1.2.2015 from address http://www.business-modelgeneration.com/downloads/business_model_canvas_poster.pdf.
- Cohn, M.(2005). *Agile estimating and planning*. New York: Pearson education inc.
- Correia, A., Abreu, F.(2009). Integrating IT Service Management within the Enterprise Architecture. *IEEE 2009 Fourth International Conference on Software Engineering Advances*.
- Deming, E. (2000). *Out of the crisis*. USA: MIT Press edition.
- Geracie, G., Eppinger, S.(2013). *The guide to the Product management and marketing Body of Knowledge (ProdBOK)*. Product Management Educational Institute.
- Henderson, J.C., Venkatraman, N.(1993). Strategic Alignment: Leveraging Information Technology for Transforming Organizations. *IBM Systems Journal*, 1993, Vol 32, NO 1.
- Hill, R., Hirsch, L., Lake, P., Moshiri, S.(2013). *Guide to Cloud Computing. Principles and practice*. Springer.
- Hype cycle for Software as a service, 2014*. (2014a). Gartner. Gartner Inc.
- Iacob, M. E., Meertens, L. O, Jonkers, H., Quartel, D. A. C., Nieuwenhuis, L. J. M., van Sinderen, M. J.(2012). *From enterprise architecture to business models and back*. Berlin Heidelberg: Springer-Verlag.
- IT Governance Institute. (2007). *CobiT 4.1*. Illinois: IT Governance Institute.
- IT Market Clock for ERP Platform Technology 2014*. (2014b). Gartner. Gartner Inc.
- Järvinen, P., Järvinen, A.(2011). *Tutkimustyön metodeista*. Tampere: Opinajan kirja.
- Järvinen, P.(2015).Pilvet karkaavat rapakon taakse.Tivi, May 2015.
- Ko, R., Lee, S., Lee, E. Business process management (BPM) standards: a survey. 2009. *Business Process Management Journal*. Vol. 15 No. 5, 2009. pp. 744-791.
- Kotler, P., Armstrong, G.(2010). *Principles of Marketing (14th edition)*. Pearson Prentice Hall: New Jersey.
- Lankhorst, M. 2012b. *Enterprise Architecture at Work. Modelling, Communication and Analysis*. Springer.

- Löhe, J., Legner, C.(2013). Overcoming implementation challenges in enterprise architecture management: a design theory for architecture-driven IT Management (AD-RIMA). *Berlin Heidelberg: Springer-Verlag*.
- Macfarlane, I., Taylor, S.(2007). *ITIL V3 Small-Scale Implementation*. United Kingdom: The Stationery Office.
- Manifesto for Agile Software Development*.(2001). Picked-up 1.2.2014 from address: <http://agilemanifesto.org>
- Meertens, L. O, Iacob, M. E., Jonkers, H., Quartel, D., Nieuwenhuis, L. J. M., van Sinderen, M. J.(2012). Mapping the Business Model Canvas to ArchiMate. *ACM SAC'12, March 25-29*, 1694 – 1701.
- Mell, P., Grance, T.(2009). *The NIST definition of cloud computing*. Version 15. Picked-up 1.3.2015 from address <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>. National Institute of Standards and Technology, Information Technology Laboratory.
- Nabiollahi, A., Alias, R. A., Sahibuddin, S.(2010). A Service Based Framework for Integration of ITIL V3 and Enterprise Architecture. *Information Technology (ITSim), 2010 International Symposium on Information Technology, Volume 1, 1 – 5*.
- Nabiollahi, A., Alias, A., Sahibuddin, S.(2011). Involvement of Service Knowledge Management System in Integration of ITIL V3 and Enterprise Architecture. *American Journal of Economics and Business Administration 3 (1)*: 165-170.
- Open Group Standard.(2011). *TOGAF Version 9.1*. Picked-up 1.2.2013 from address: <http://pubs.opengroup.org/architecture/togaf9-doc/arch/>.
- Osterwalder, A.(2004). *The business model ontology. A proposition in a design science approach*. Thesis: Université de Lausanne.
- Osterwalder, A., Pigneur, Y. (2010). *Business Model Generation*. John Wiley & Sons.
- Peffer, K., Tuunanen, T., Rothenberger, M., Chatterjee, S., A Design science research methodology for Information systems research. 2007. *Journal of Management Information Systems*, Winter 2007-8, Vol. 24, No. 3, pp 45-77.
- Peppard, J., Ward, J., Daniel, E.(2007). Managing the realization of business benefits from IT investments. *MIS Quarterly Executive*, Vol. 6 No. 1.
- Porter, M.(2001). *Strategy and the Internet*. Harvard Business Review, March, 2001.
- Ross, J.(2005). Forget strategy: focus IT on your operating model. *CISR Research Briefing*, Vol. V, No. 3C. December 2005.
- Saarelainen, A.(2015).Nokian taival vie pilveen. Tivi, May 2015.
- Shpilberg, D., Berez, S., Puryear, R., Shah, S.(2007). Avoiding the alignment trap in information technology. *MIT Sloan management review*. Fall 2007.
- Siltala, T.(2015).Isompi siivu verkosta. Tivi, May 2015.
- Sun, Y., Xiao, Z., Bao, D., Zhao, J. (2010). An Architecture Model of Management and Monitoring on Cloud Services Resources. *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*.
- Valiente, M-C., García-Barriocanal, E., Sicilia M-A.(2011). Applying Ontology-Based Models for Supporting Integrated Software Development and IT Service Management Processes. *IEEE Transactions On Systems, Man, And Cybernetics, Part C: Applications And Reviews*, Vol. 42, 61 – 74.

- Wand, Y., Weber, R.(1993). On the ontological expressiveness of information systems analysis and design grammars. *Information Systems Journal*, vol. 3, no. 4, 217-237.
- Wand, Y. (1996). Ontology as a foundation for meta-modelling and method engineering. *Information and software Technology* 38, 281 – 287.
- Wand, Y., Storey, V., Weber, R.(2000). An Ontological Analysis of the Relationship Construct in Conceptual Modeling. *ACM Transactions on Database Systems*, Vol. 24, No. 4, 494–528.
- Zachman, J.(1987). A framework for information systems architecture. *IBM Systems Journal*, Vol26, No 3, 454 – 470.
- Zachman, J.(2000). *Conceptual, Logical, Physical: It Is Simple*. Picked up 1.4.2013 from address: <http://www.zachman.com/ea-articles-reference/58-conceptual-logical-physical-it-is-simple-by-john-a-zachman>.
- Zachman, J.(2011). *The Zachman framework for enterprice Architecture*. Version 3.0. Picked-up 1.4.2013 from address: <http://www.zachman.com>.

ATTACHMENTS

Attachment 1: IT Service Management frameworks ontology analysis results

If either framework is missing the process relative column is left empty. Ontological problem is defined on the right column.

CobiT processes	ITIL processes	Ontology problem
PO1 Define a Strategic IT Plan	Strategy Generation	
	Demand Management	Missing
PO2 Define the Information Architecture		Missing
	Service Portfolio Management	Missing
PO3 Determine Technological Direction		Missing
PO4 Define the IT Processes, Organisation and Relationships		Missing
PO5 Manage the IT Investment	Financial Management	
PO6 Communicate Management Aims and Direction		Missing
PO7 Manage IT Human Resources		Missing
PO8 Manage Quality		Missing
PO9 Assess and Manage IT Risks		Missing
PO10 Manage Projects		Missing
AI1 Identify Automated Solutions		Missing
AI2 Acquire and Maintain Application Software		Missing
AI3 Acquire and Maintain Technology Infrastructure		Missing
AI4 Enable Operation and Use		Missing
AI5 Procure IT Resources	Supplier Management	
AI6 Manage Changes	Change management	
AI7 Install and Accredit Solutions and Changes		Missing

CobiT processes	ITIL processes	Ontology problem
DS1 Define and Manage Service Levels	Service Level Management	
	Catalogue Management	Missing
DS2 Manage Third-party Services		Missing
DS3 Manage Performance and Capacity	Capacity Management	
DS4 Ensure Continuous Service	Service Continuity Management	
DS5 Ensure Systems Security	Information Security Management	
	Transition Planning and Support	Missing
DS6 Identify and Allocate Costs		Missing
DS7 Educate and Train Users		Missing
DS8 Manage Service Desk and Incidents	Incident Management Event Management Request Fulfilment	Overload
DS9 Manage the Configuration	Service asset and configuration management	
DS10 Manage Problems	Problem Management	
DS11 Manage Data		Missing
DS12 Manage the Physical Environment		Missing
DS13 Manage Operations	Operation Management	
	Availability Management	Missing
ME1 Monitor and Evaluate IT Performance	Service Measurement Service Reporting	Overload
ME2 Monitor and Evaluate Internal Control		Missing

CobiT processes	ITIL processes	Ontology problem
ME3 Ensure Compliance With External Requirements		Missing
ME4 Provide IT Governance		Missing
	Release and Deployment Management	Missing
	Service validation and testing	Missing
	Evaluation	Missing
	Knowledge Management	Missing
	Service Improvement	Missing

Attachment 2: Ontological relations of EA and ITSM frameworks

Only relevant processes from CobiT and ITIL frameworks are shown. If process doesn't map with TOGAF domains it's omitted.

TOGAF domains	Zachman views	Cobit framework	ITIL framework
Business architecture	Scope contexts Business concepts	PO1 Define a Strategic IT Plan PO4 Define the IT Processes, Organisation and Relationships PO5 Manage the IT Investment PO6 Communicate Management Aims and Direction PO7 Manage IT Human Resources PO8 Manage Quality PO9 Assess and Manage IT Risks PO10 Manage Projects AI5 Procure IT Resources DS1 Define and Manage Service Levels DS2 Manage Third-party Services DS3 Manage Performance and Capacity DS4 Ensure Continuous Service DS5 Ensure Systems Security DS6 Identify and Allocate Costs DS7 Educate and Train Users DS8 Manage Service Desk and Incidents DS10 Manage Problems DS13 Manage Operations ME1 Monitor and Evaluate IT Performance ME2 Monitor and Evaluate Internal Control ME3 Ensure Compliance With External Requirements ME4 Provide IT Governance	Service Improvement Service Measurement Service Reporting Operation Management Problem Management Incident Management Change management Event Management Request Fullfilment Transition Planning and Support Information Security Management Service Continuity Management Financial Management Service Portfolio Management Demand Management Strategy Generation

TOGAF domains	Zachman views	Cobit framework	ITIL framework
Data architecture		DS11 Manage Data	Knowledge Management
Application architecture	System logic Technology physics	DS9 Manage the Configuration AI2 Acquire and Maintain Application Software AI1 Identify Automated Solutions AI6 Manage Changes AI7 Install and Accredite Solutions and Changes	Service asset and configuration management Release and Deployment Management Service validation and testing Evaluation Service Improvement
Technology architecture	Tool components Operations instances	AI3 Acquire and Maintain Technology Infrastructure	Availability Management Capacity Management Catalogue Management Service Level Management Supplier Management

Attachment 3: Capability assessment template

Capabilities which are taken from ontological analysis of EA and ITSM frameworks are highlighted (bolded) in the capability assessment template. ID defines the capability identifier in the capability assessment tool, name is the default name and definition is the capability definition from the product viewpoint.

Default business capabilities

ID	Name	Definition
B1	Marketing model	How product is marketed for target customers, channels, activities, resources needed
B2	Sales model	How product is sold, channels, activities, resources needed
B3	Delivery model	How product is delivered for the customer, order to delivery model, resources needed
B4	Support model	How customers are supported, channels, activities, resources and availability needed
B5	Operations model	How operations are managed, resources and availability needed
B6	Service reporting	How service is measured and reported to customer (SLA)
B7	Incident management	How customer incidents are managed, target response and resolution times
B8	Problem management	How customer problems are managed, target response and resolution times
B9	Change management	How customer change requests are managed, target response and resolution times
B10	Financial management	How costs and incomes are managed inside organisation, customer invoicing, customer disputes

Default data capabilities

ID	Name	Definition
D1	Marketing material	Web sites, brochures etc
D2	Sales material	Presentations etc
D3	Pricing	Pricing lists
D4	Contract material	Customer contract
D5	Instructions	Help and how to use the product
D6	Knowledge management	How knowledge is transferred to stakeholders inside organisation in order to provide needed capabilities, where product knowledgebase is kept, who has access to it

Default application capabilities

ID	Name	Definition
A1	Tenant provisioning	How tenant is provisioned
A2	Backend integrations	Integrations to company backend systems
A3	Release and deployment management	How updates are released from customer point of view

Default technology capabilities

ID	Name	Definition
T1	Supplier management	Who are suppliers and how they are managed, specific requirements for supplier
T2	Capacity management	How much capacity is needed and how it's managed
T3	Service level management	How service levels are measured and reported
T4	Service asset and configuration management	How service assets and configurations are managed
T5	Technology stack	Is technology stack defined, technology vendors known and managed, licences acquired
T6	Development infrastructure	Development infrastructure needed
T7	Testing infrastructure	Testing infrastructure needed
T8	Production infrastructure	Production infrastructure needed

Attachment 4: Cloud product capability assessment toolkit

Method and tools development is continuous activity like explained in the DSRM process. Also important part of the design science is to make methods available and acquire more feedback from academia and other professionals by utilising the method in different context. In order to support continuous development of the cloud product capability assessment method tools and instructions mentioned in this thesis are made available as open source toolkit.

Cloud product capability assessment toolkit includes following items which were developed during the research process:

- Cloud product capability assessment guidebook
- Planning poker cards
- Thesis
- The assessment tool

The assessment tool was developed with OpenOffice 4.0.1 version. More information about OpenOffice: <http://www.openoffice.org>. Future versions of the cloud product capability assessment toolkit can include support for other OpenOffice versions as well as possible other productivity tools, like Microsoft Office. These are done based on the feedback and need from the users.

Download and give feedback for cloud product capability assessment toolkit: <http://cloudproductassessmenttoolkit.sourceforge.net>

Attachment 5: Live events example case

Product business model

Product: Live events		Version	0.3
		Last update	01.03.15
Key partners <i>Who are our Key Partners? Who are our key suppliers? Which Key Resources are we acquiring from partners? Which Key Activities do partners perform?</i>	Key activities <i>What Key Activities do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue streams?</i>	Value proposition <i>What value do we deliver to the customer? Which one of our customer's problems are we helping to solve? What bundles of products and services are we offering to each Customer Segment? Which customer needs are we satisfying?</i>	Customer relationship <i>What type of relationship does each of our Customer Segments expect us to establish and maintain with them? Which ones have we established? How are they integrated with the rest of our business model? How costly are they?</i>
Event suppliers - has actors, competencies and facilities to organize events Program offices - manage bands, artists and teams Consumers - Organize personal events - Buy tickets for events Credit card companies - Easy way to pay services and tickets	Platform development and operations Network management Sales & Delivery Key resources <i>What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue Streams?</i> Platform development (R&D) Platform operations (support & maintenance) Network management Sales & Delivery	Simple access of extensive global network of suppliers who can organize any kind of events for enterprises. Enterprise can extend own network with Live events product, professional services and suppliers. Easy to find and simple to buy tickets for local events for consumers. Simple way to organize personal events like birthdays. Better visibility and more business for event suppliers.	Enterprises / Program offices - Simple events automated - Complex events tailored Consumers - Automated, no personal assistance anything else than ticket transactions Channels <i>Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones work best? Which ones are most cost-efficient? How are we integrating them with customer routines?</i> Enterprises / Program offices - Simple events: Mostly online channel - Complex events: Online + F2F - Supplier network: mostly online Consumers - Online channel + support line for ticket transaction problems
Cost structure <i>What are the most important costs inherent in our business model? Which Key Resources are most expensive? Which Key Activities are most expensive?</i>		Revenue streams <i>For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How would they prefer to pay? How much does each Revenue Stream contribute to overall revenues?</i>	
Platform costs (development + operations) Cost of sales for enterprises (customer acquisition costs) & delivery of events Network management (nurturing suppliers & program offices to join to the network)		Ticket commissions (10%) Even management & organizing (90%)	

Assessment status after the first planning workshop

Capability assessment: Live events						
ID	Capability	Description	Investment type	Scope	CPCI	Status
Total	33	33	15	9	9	12

Assessment status

Category	Value
Capability	33
Description	33
Investment type	15
Scope	9
CPCI	9
Status	12

Kanban board status after the first planning workshop

Product: Live events							
Development Kanban							
ID	Capability	Definition of done	Priority	Owner	Open (10)	WIP (0)	Done (0)
T1	Supplier management	Suppliers evaluated and selected		1 David	6		
T5	Technical feasibility	Technical risks understood, competencies for development defined		1 David	16		
A1	B2B registration	Acceptance testing done		2 Lisa	26		
A5	Consumer registration	Acceptance testing done		2 Lisa	6		
A3	Release and deployment management	Acceptance testing done		3 Lisa	26		
A8	Event management	Acceptance testing done		3 Lisa	80		
A4	Credit card integration	Acceptance testing done		3 Lisa	40		
A2	Backend integrations	Acceptance testing done		4 Lisa	12		

Attachment 6: Survey questions

#	Question	Answers	Reasoning
1	Select your unit or team	Product management R&D Support Delivery Operations Other, please specify	Background information
2	Select your job title / responsibility area	Product manager System analyst / product owner Architect Project manager Team leader Developer / Tester Specialist / Engineer Other, please specify	Background information
3	Select which session you participated	Interview Group workshop Both	Background information
4	Workshop targets were clearly presented	Likert scale	Analyze if workshop target was clearly described and defined
5	Group assessment (planning poker with cards) method was easy to understand	Likert scale	Analyze if group assessment (planning poker) method was clear and easy to understand
6	Discussion around product business model helped me to understand product vision	Likert scale	Analyze if business model canvas helps to understand big picture of product
7	After workshop I understand what a capability is and what kind of different capabilities are needed to bring product to market.	Likert scale	Analyze if workshop was beneficial to understand what product development holistically needs
8	Scope analysis with cards and discussion with others helped me to understand expectations and development complexities	Likert scale	Analyze if relative analysis of capabilities and discussion with other stakeholders and colleagues makes it easier to understand development complexities
9	Assessment status was clearly presented	Likert scale	Analyze whether assessment status was clearly visualized
10	Product development status was clearly presented	Likert scale	Analyze if development status was clearly visualized
11	Individual capability scope and status was clearly presented	Likert scale	Analyze if development item visualization helps to understand different development items relative size (scope)
12	For me the workshop and group assessment was	Useless to Useful	Analyze if person thinks workshop was overall beneficial
13	Future workshops would be	waste of my time to help to commit to product development	Analyze if person wants to use the method and participate continuous workshops in the future.
14	What worked well	Open question	Get open feedback what worked
15	What should be improved	Open question	Get open feedback what should be improved