MEASUREMENT PRACTICES, METRICS AND MEASURES FOR ASSESSING THE VALUE OF ENTERPRISE LEVEL AGILE IT PROJECTS

MASTER OF SCIENCE THESIS

LIANA MARIANA UILECAN

FEBRUARY, 2017

MEASUREMENT PRACTICES, METRICS AND MEASURES FOR ASSESSING THE VALUE OF ENTERPRISE LEVEL AGILE IT PROJECTS

Thesis submitted to

Delft University of Technology

In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in Management of Technology

Faculty of Technology, Policy and Management

By

Liana Mariana Uilecan

Student Number: 4401742

Graduation Committee

Chair: Prof. Dr. Ir. Marijn Janssen

First Supervisor: Drs. Jolien Ubacht

Second Supervisor: Dr. Martijn Warnier

External Supervisor: Erwin Mul, Shell

This page was intentionally left blank

ACKNOWLEDGEMENTS

The conclusion of my master studies lies in this thesis. The journey to reach this final step was filled with challenging but at the same time fulfilling experiences. This process was made easier and more rewarding with the help and support of people surrounding me. Thus, I am taking this moment to express my sincerest gratitude to them.

I would like to express my deepest gratitude to my first supervisor, Jolien Ubacht, for her patience, motivation and guidance. During the first part of the research she gave me confidence in shaping my own research study and narrowing the scope. Of course, throughout the months that followed her continuous feedback and support were unconditional. I would like to also thank my thesis committee, the chair, Professor Marijn Janssen, and second supervisor Martijn Warnier for their insightful and challenging feedback and encouragement as well.

Moreover, my sincerest thanks to Erwin Mul, my Shell mentor who gave me the opportunity to join his team as an intern. Without his support and access to internal network of experts this research would not have been possible.

Finally, I want to thank my family and brother for all their encouragement and invaluable support. I took your advice at a crossroad and now I have a fancy Master degree to thank you for. And to my number one supporter and my rock, Tudor, the darkness has not won.

Liana Uilecan

February, 2017

This page was intentionally left blank

EXTENDED ABSTRACT

The adoption of agile software development methodologies is still on the rise (Olszewska, 2016) with more and more organizations using this methodology to attain the objectives of decreasing cycle times and increasing value of software releases to customers (Dymond, 2006; Dingsøyr, 2012). A reason for which agile software development is still on the rise, even after more than a decade since its apparition (Beck, 2001), is the increasing trend of IT insourcing (TechTarget.com, 2013). IT functions, including in-house software development, are insourced more and more especially amongst large companies, who require higher control over IT technology expertise to leverage fast new product development and increased speed to market (Shacklett, 2012). However, due to this only recent shift from outsourcing, large companies with core businesses other than software development seldom have agile experience in-house. With agile being originally built for small and relatively independent development teams, the complex organizational and governance structures which involve multiple teams and hierarchies pose challenges for large companies to adopt agile (Laanti, 2008).

To monitor and attain the objectives of agile software delivery and surpass its challenges within a corporate environment, agile project managers and owners need to measure the value of the end software product compared to quality objectives. This is achieved through software measurement, a branch of software quality management which purposes to establish and guide the application of a framework of processes and standards that lead to high quality software (Sommerville, 2012). Traditionally, within plan-driven work methodologies, software measurement focuses mostly on efficiency-oriented methods, that gauge software performance, project costs or productivity (Jones, 2008; Zuse, 1998) rather than value delivered to client as top priority quality goals.

However, due to the shift from plan-driven to agile, historical software measurement methodologies must also change to track and attain the goals of agile software. Because agile and plan-driven approaches impose different evaluation methods, utilizing plan-driven measurement in agile development proves to be not only ineffective, but counter-productive (Morasca, 2001; Hartmann, 2006). Agile measurement should indicate quality goals of software such as its adoption rates, the quickness of delivery, and the flexibility of change in requirements (Boehm, 2005; Sommerville, 2012). Thus, a core challenge for scaling agile to larger organizations is adapting measurement practices to accurately assess and ensure quality goals (Sommerville, 2012).

Despite the recognized importance of assessing software value (Biffl, 2006; Jones, 2008; Sommerville, 2012), agile measurement programs have mostly been unsuccessful, being done incompletely through superficial client satisfaction surveys (Jones, 2008), or incorrectly by applying plan-driven measurement frameworks (Hartmann, 2006; Kulas, 2012). Thus, we identified a gap in theory and practice when discussing about agile measurement frameworks that include metrics, measures and practices that can gauge the value of agile projects (Javdani, 2013). Some of the reasons for why this gap still exists, despite agile being an already mature software development practice, are the dynamic nature of requirements and practices in Agile (Javdani, 2013), and the difficulty to quantify quality objectives such as fulfilling customer needs and business value (Fehlmann, 2014). However, there are many disparate research articles and sources that describe measures that may be used for agile software product value

assessment (Dale, 1992; Petersen, 2010; Olszewska, 2016; Hartmann, 2006; Marciuska, 2014; Herzog, 2013; Favaro, 2003), although it is not clear how they would apply in a large company environment, and how they can be combined together into a framework.

Situated within the setting of Shell's transformation program towards scaled agile, the objective of this thesis is to propose a set of metrics, measures and recommendations, targeted at IT project stakeholders to assess the value of agile software development projects in an enterprise context.

In order to achieve this objective, several research approaches were applied. An initial literature review was conducted to set the context of why companies are increasingly utilizing agile software development methodologies compared to plan-driven methodologies, their measurement approaches, the benefits and challenges of applying a measurement framework, and the focus of value as a key quality objective for developing software through agile methodologies. Through this stage, we found that there is a severe lack of consistency of applying agile measurement due to the absence of a measurement framework, forcing companies to misuse and underappreciate measurement. Upon defining the problem and its scientific and practical relevance, we set out to split it into several research questions and sub questions which needed different approaches to be answered.

First, by conducting a more in-depth literature review of scientific articles on software measurement of plan-driven and agile development projects, we reinforced the argument that existing literature scarcely treats agile measurement through value-oriented metrics and measures. We did this by underlining that most articles on measurement focus on project efficiency and effectiveness rather than value, but at the same time we found compelling evidence that these methodologies should not be applied for agile development without being complemented by a majority of value metrics. We then identified and extracted a series of metrics and measurement practices that can be combined with efficiency and effectiveness metrics to build an agile value-measurement framework. We categorized the value metrics as being targeted at measuring product usage, uptake or client satisfaction, proposing that a relatively balanced combination of the three is needed to assess overall value.

Second, to support the theoretical assumptions built through literature review and analysis, we set out to conduct three semi-structured interviews with Shell employees who were involved in IT projects and had experience with plan-driven and agile project management (HR IT portfolio manager, IT solutions architect and IT Product Owner). The semi-structured interviews helped explore the possibilities of utilizing agile software measurement practices, who the interviewees had little experience with, but with which they were in strong agreement that they are necessary for IT quality management and agile development. The limitations faced in the data collection phase are related to the scarcity of expert knowledge on measurement practices (especially with respect to agile projects) and the relatively few number of interview respondents. In the next phase, we analyzed the responses and pitted them against our theoretical findings in order to construct the final agile value measurement framework.

Third, the data analysis of empirical data was made by coding and categorizing data into (1) "*metrics that influence business/financial measurement*" for identifying the link between value measurement and the conclusions for business stakeholders, (2) "*decision making based on metrics*" for understanding what actionable outcomes can result out of monitoring and analysis of metrics, (3) *"agile vs. waterfall measurement"* for judging the differences and similarities of applying value measurement to the two

work methodologies, and lastly, (4) "*challenges and solutions for measurement"* a unit reflecting on the core challenges of applying a measurement program and potential solutions to these challenges. This final category proposes how measures and metrics can be applied in a practical context.

We found that most metrics, measures and measurement practices were not new to the respondents and even that these metrics could be collected from a technical perspective, but that they were unused because of the difficulty to interpret them utilize them as a relevant basis for decision making. Moreover, most challenging side of implementing a measurement program is not necessarily in the choice of the metrics and measures, but in setting up and maintaining the right context throughout development and operation of software. Based on these findings, we recommended that a value measurement program should be operated simultaneously with agile development and operation stages, and that stakeholders should have clear measurement responsibilities linked to their application management responsibilities. We presented a few high-level steps of a value measurement program should be linked with development, and described how delivery, operations and business stakeholders should collaborate in the planning, implementation and continuous adaptation of value measurement, to fit with project needs and characteristics, as it progresses in its application lifecycle.

Finally, because the measurement framework and recommendations could not be tested in an actual use-case scenario, we conducted an evaluation characterized by self-reflection and comparison against research quality criteria. The ex-ante evaluation of the thesis results and methodology was made from both practical and research perspectives concluding that: (1) purposiveness of research objectives and the rigor and objectivity of literature review and findings phase is relatively high, due to the selection of a wide range of articles that remain neutral and factual with respect to the benefits and challenges of agile; (2) the data collection methodology on the one hand was appropriate and rigorous because the semi-structured interviews fitted well with the relatively explorative topics of discussion, but on the other hand the choice and number of interview respondents could have been improved, to provide better insights into the specific topic of agile measurement; (3) the research results can be said to have low generalizability due to a low number of respondents combined with the fact that there may be various contexts in which measurement will differ. This supports the fact already discussed in literature that a silver bullet solution, i.e. a framework which directly solves how measurement should be done, is unlikely to exist due to the very different particularities and context factors of agile development; we consider high reproducibility because the literature findings were thoroughly referenced and the interview respondents' profiles and issues are not company-specific but can be identified in any similar sized company that runs agile software methodologies and measurement projects. Finally (4) the practical use and importance of such a framework has been well received by interview respondents, but on the downside, there are still unclarities on how implementation would happen with respect to what kind of roles would be ideal for measurement and what processes would be used to arrive from raw data collected through measurement to actionable conclusions that can help decision making.

Thus, the results of this research set the basis for further studies into both depth and breadth on how to apply agile software measurement in large companies. On the one hand, applied case studies would be needed to see how the framework could actually be applied and what results it would hold in a particular agile project. On the other hand, perhaps the more difficult task would be to further attempt

to refine the framework's broad application for agile measurement, defining what roles, processes and governance structures can be used for most agile projects, as a more concrete starting point for developing specific measurement strategies for an agile development portfolio.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 INTRODUCTION

1.1 Introduction

Throughout this introductory chapter of the research, key areas will be discussed. First, the situation and research context is introduced in Section 1.2. Second, the research problem and objective is described Section 1.3. Third, in Section 1.4 the societal and scientific relevance is presented. Next, the research questions are defined in Section 1.5 and Section 1.6 contains the research approach to study the topic of agile measurement. The chapter ends with the thesis outline in Section 1.7.

1.2 Situation / Research Context

In this section, we explain the context in which the research and practical problem are studied. We describe the domains of IT insourcing and business alignment and their relationship with agile software development in large organizations. We briefly describe the situation and core challenges of scaling agile on organizational, multi-project level and introduce the practical situation at Shell. This situation will help introduce the problem - the lack of consistent measurements systems for agile work methodologies in the case of multi-projects environments (in our case, Shell's IT organization and the transformations they undergo towards agile).

1.2.1 Insourcing of IT

Market studies indicate that IT insourcing trends are increasing, either because of unsatisfactory outsourcing contracts or due to a desire to increase control over core functions and processes, costs, and customer satisfaction. (TechTarget.com, 2013). The need for agility also pushes towards in-sourcing, long-term contracts and slow communication with outsourcers are not viable anymore for products that require fast times to market. Perhaps the most surprising reason for shifting from outsourcing to insourcing is cutting costs (which used to be a prime objective of outsourcing). A survey from Deloitte showed 77% of respondents choosing to bring back IT functions in-house for cost cutting reasons (TechTarget.com, 2013).

Extensive levels of organizational IT outsourcing not only pose serious issues of control but erode internal IT capabilities of companies. IT insourcing decisions are increasingly common amongst large companies, as the need for control over IT technology expertise, new product development, speed to market, and future IT capability are on the rise (Shacklett, 2012). IT insourcing is in some cases more effective for developing IT-enabled business processes, subsequently leading to superior company performance. This means that companies need to consider IT as an integral component of their strategy and continuously develop internal IT resources (Qu, 2010).

1.2.2 Hybridization of IT (business-IT alignment)

Companies' IT departments switch from IT deliverers to end-to-end providers of services and solutions. IT departments must nowadays provide IT-as-a-service for the internal company. Failing to change

quickly or effectively enough in this direction causes organizations to bypass their internal IT and consume external services. (Hall, 2011).

For several decades, IT and business have been increasingly intertwined, with IT being deeply integrated within business processes. This concept is largely known in literature as business-IT alignment: the evolutionary and dynamic process of IT and business functions adapting to each other and following joint strategies (Luftman, 2004). This idea may come under many forms, practitioners discussing about alignment interchangeably with harmony, linkage, fusion, integration or hybridization.

In some instances, the IT-business alignment is referred to as hybridization of IT - the involvement of business resources in IT delivery and vice versa, empowered by cross-disciplinary skillsets of both business and IT (Source: Shell strategy). One facet of the difficult challenges faced when IT hybridization is underway applies to IT and business managers, who must work together on how to integrate IT in business processes and routines. Moreover, the line between IT and business is blurred even for R&D purposes, where IT-based innovation should no longer be delegated to IT management but managed in a coordinated effort by cross-departmental managers (Peterson, 2000).

1.2.3 Scaled Agile Framework

As underlined in the previous subsections, many companies are looking to improve their internal business processes and external delivered value means of IT. Insourcing and IT-business alignment increase control over company-grown capabilities, IT project effectiveness, delivery times and quality, and fit between business demand and IT supply. However, insourcing IT implies that projects must be delivered with at least the same quality and speed as outsourced projects would be. In the volatile business and technology environment of today, insourcing calls for IT project development and delivery methodologies that are flexible towards changes in customer requirements and organizational constraints. (Highsmith, 2001)

A suite of development methodologies that supports the needs enumerated above is agile methodologies. Since the materialization of agile principles in the “Agile manifesto” in 2001 (Fowler, 2001), agile software development methods adopted by IT teams has arguably increased delivered value of software to clients (Dingsøyr, 2012). Agile software teams encourage customer involvement in the development process and can accommodate changes in client requirements at any time during development. This new way of working brings out value from insourcing and facilitates IT-business alignment through its fast and adaptable development cycles (sprints) and close collaboration with business stakeholders.

However, agile methodologies were designed to apply on the software development process level but not on whole project portfolios or even organizations. Where multiple teams and projects are involved, deploying agile methods requires coordinated decision-making, communication between project teams and managers and synchronization with other activities (Laanti, 2008). Large organizations can reap the benefits of agile while maintaining appropriate levels of control and consistent decision making if the whole organization is transformed, instead of having disparate, unlinked clusters of teams working agile (Laanti, 2013). The application of agile on organizational level for large businesses has been touched in

research literature (Laanti, 2013) and practice such as the Scaled Agile Framework (SAFe), or Disciplined agile delivery (DAD) but this area is largely still under research (Leffingwell, 2011; Ambler, 2012).

To track how agile work methodologies compare to traditional ways of working, and if they can deliver value in the midst of enterprise transformations towards IT insourcing and tighter business alignment, measurement is needed. This will be the focus of the next subsection where the problem is treated. At this point, we mention that this study revolves around the case of Shell's IT transformation from traditional to more agile-oriented development methods. Shell IT is currently implementing an IT strategy strongly focused on insourcing and business-IT hybridization, with the Scaled Agile Framework as one facet of change. SAFe is being applied to support an organization wide shift of IT towards agile methodologies where they are appropriate. A major issue in this transformation is keeping track of what projects and teams produce higher value with agile, and how to measure and compare these results on project and portfolio levels (with each other and with old ways of working such as waterfall). We will describe the Shell problem context in addition to the research problem context in the following subsection.

1.3 Problem

Measuring the value, cost, quality and efficiency of IT project deliverables is a challenging and often discounted part of organizational IT management, especially when it comes to measuring effectiveness of the solution with respect to business needs. Traditionally, the measurement of IT project value and performance has been approached from a technical, efficiency-oriented perspective (Jones, 2008). Efficiency oriented measurements measure performance of software, project costs, productivity and other factors (Jones, 2008; Zuse, 1998) which provide insights into the resource costs (input) of projects, but are weaker in reflecting business value (output). With the increasing presence and influence of IT in business, a higher degree of integration between the two is needed for IT to deliver value. From a performance measurement perspective, the increased Business-IT hybridization of projects calls for broader, more comprehensive practices for measuring effectiveness. Ideally, effectiveness measurements should reflect the value of delivered projects through metrics such as adoption rate, utilization, customer satisfaction, etc. This approach can be extremely valuable for decision makers to correctly allocate resources, and manage increasingly complex and volatile IT portfolios. However, the ideal case is underdeveloped in practice - performance measurements are mostly efficiency and technical-oriented, meaning that business stakeholders rarely receive any metrics relevant to them.

In the case of Shell, business and IT managers feel the need to complement existing performance measurement methods with value measurement, and utilize a consistent method across different projects and teams within the IT function. In the new project development and delivery setup described in the situation section, Shell expects to deliver fewer 'units of work' overall while at the same time delivering them earlier and in a more targeted fashion. Following agile work methodologies and continuous delivery practices, they expect these 'units' to be of higher value than those delivered in pure-waterfall IT projects. However, most IT performance measurements focus on the 'efficiency of delivery' such as function point per man hour, technical quality 'uptime' or 'defect rate'. While these metrics measure efficiency, IT managers do not have a standard way to measure value through metrics

such as dimensions of utilization, satisfaction, bottom line value delivered, etc. At the moment, value is mostly assessed via ad-hoc practices such as learnings of past experiences, input gathered from different project stakeholders, etc. These methods reduce the effectiveness of continuous improvement practices (Gartner), thereby undermining the benefits of new development methods that are implemented at Shell.

Therefore, the problem can be phrased as:

There is no consistent tool or methodology to measure the effectiveness and business value of hybrid IT-business projects and portfolios within the Scaled Agile framework.

To reinforce the importance of such a methodology, we underline core consequences of lacking a coherent and transparent measurement framework. First, longitudinal and cross-sectional benchmarking and comparison of internal projects' performance cannot be done if different metrics and practices are utilized for measurement and reporting. Second, strategic decisions cannot be made with incomplete or inaccurate data. Third, existing measurement practices' costs and effectiveness are hard to be assessed due to difficulty of capturing, storing and transferring knowledge that is non-overlapping, non-redundant and reusable.

Multiple problem owners can be identified because the entire measurement process affects different functional and hierarchical levels. Stakeholders should communicate in the decision-making process for choosing the measurement goals, methods and evaluation. (1) The CIO sets strategic objectives and defines the role of IT measurement within these objectives, as well as champions high-level measurement initiatives (ensures business and financial support for the IT initiatives). (2) Portfolio managers are tasked with implementing the strategy - choosing how measurements are carried out in a consistent, standard way across projects, evaluating results to compare different projects and the portfolio as a whole. (3) Project managers are responsible for implementing and monitoring that the measurement processes are followed on operational level, on individual projects. They are also responsible for managing change, discussing with development project team leaders about what measures are should be used and how they should be implemented, and ensuring that the teams are satisfied and have the necessary resource to execute the additional measurement tasks. Finally, project managers round up and interpret result reports and communicate them to portfolio managers.

The research objective is:

> ***To deliver a set of metrics, measures and recommendatios, targeted at IT project stakeholders to assess the value of agile software development projects in and enterprise context.***

The set of metrics, measures and recommendations, which we call "framework" thus forward for simplicity, should be robust enough to be integrated within broader IT management frameworks (ITIL, Cobit, etc.) and complement other measurement and monitoring tools. One of the requirements of the framework is that its form should be as simple as a form / excel sheet that can be filled by different IT project stakeholders. Further requirements will be explored during the research project.

1.4 Study Relevance

In this section, an overview of the main incentives to conduct this research is given. The relevance of this study is split between academia and practitioner domains. The following sections motivate the scientific relevance of the thesis by describing the literature gap and how it is addressed by delivering a measurement framework for agile IT project effectiveness and value measurement. The societal or practitioner relevance is explained through the lens of how the deliverable can be applied in IT organization contexts to help companies measure the value of projects working with agile methodologies within a scaled agile framework.

1.4.1 Scientific relevance

Currently, the value of agile IT projects is mostly measured in terms of efficiency, using indicators such as delivery times, defects, costs and others. The value of software and implicitly of IT departments has been recognized as difficult to assess, but much needed especially in the context of increased alignment between business and IT (Biffl, 2006). However, effectiveness and business value of the project and final deliverable is most of the times superficially measured through client satisfaction surveys (Jones, 2008). In the context of scaled agile being applied in large IT organizations, measures of project value and effectiveness are essential for decision makers that choose the IT portfolio composition. Literature lacks a consistent measurement framework that specifically underlines the value of agile IT projects and work methodologies, despite the existence of many software measurement methods, frameworks and indicators that address efficiency and quality (Measure Defect-Density, lines-of-code, Function-Point Method, etc.) (Zuse, 1998).

Measurement of agile software development is mostly treated in practitioner literature (expert magazines and conferences, blogs and websites), with most methods and metrics applied on agile being inappropriate because of their belonging to classical software development (Hartmann, 2006). Therefore, a clear research gap can be delineated in measuring agile projects with appropriate measures. We address this gap by filling the specific area in agile measurement which relates to value and effectiveness assessment - indicators which are much needed in the context of scaled agile (SAFe) and multiple projects that form IT portfolios rather than individual projects. This research proposes a new perspective on measuring the value of IT projects, throughout the Software Development Life Cycle (SDLC), by focusing on quantitative measurements of effectiveness and business value of IT projects within IT portfolios.

Some examples of indicators for measuring value found in literature are (1) compliance to the process, (2) functionality adoption (utilization/penetration) adoption rate / usage patterns, (3) process coverage and (4) customer satisfaction.

These metrics and others have been researched in detail throughout the study at hand. Our aim is not to replace, but renew existing methods with added metrics, into a comprehensive framework, complementing traditional efficiency measurement methods with agile-specific measurements that indicate IT project value. An additional characteristic that agile-specific measurements supports (and is novel compared to purely waterfall projects) is the continuous data generation from measurements that are used during development within a feedback loop that improves the deliverable, before it is released.

IT project managers can act upon the knowledge provided by value measurements taken during development (or during other non-final steps of the development lifecycle) so that the project can be adapted and improved, coming closer to the value expected by business and clients. Moreover, at a higher level, portfolio managers can decide to continue or discontinue a project while still under development if project level indicators show evidence that the project is or is not valuable (when compared to requirements, or when compared to the value of other projects in the portfolio).

1.4.2 Societal relevance

The impact of our research in practice is mostly targeted at IT and business decision makers that govern the IT organization in large companies with multiple IT projects that are conducted under mixed ways of working (waterfall and agile mix). Our research could provide a basis for planning and implementing a measurement strategy to assess the value of agile IT projects, or to evaluate projects that transition from classical (waterfall) to novel (agile) methodologies. IT project and portfolio decision makers may use project-specific metrics to estimate KPIs at both project and portfolio levels, thereby contributing to IT project decisions related to: which projects should be undertaken, which areas benefit from agile methodologies, and so on[1]. The need of a consistent measurement framework that can be used in a standard way across different project teams exists in practice, as we emphasize in the case study of Shell's IT organization. Despite the fact that our will study scientific and practitioner literature with general applicability on agile IT projects, the interview and workshop parts will happen within Shell and be addressed to Shell stakeholders. This latter part may reduce the generalizability of our findings on other companies and cases. However, our intention is to develop a framework that can be applied in other organizations, assuming that they also involve IT transformation from classical to agile work methodologies in the context of multiple IT projects and complex IT organizations (large, IT intensive companies).

1.5 Research Questions

The current section provides the research questions guiding the research process in solving the problem described above.

1.5.1 Main research question

The following main research question needs to be answered in order to deliver the measurement framework:

How would a value measurement framework assisting project stakeholders in identifying benefits and challenges of scaled agile IT projects look like?

[1] There is a large extent of project types that dictate the consideration of different KPI requirements. For example, projects can be classified after team structure (cross-departmental, cross-organizational, international, virtual, etc.), deliverable type (internal use, external clients, supporting other applications, etc.). Further in our thesis we will explain what KPIs / metrics are suitable to be collected depending on the goals of different types of projects.

The answer to this research question will enable business managers to make informed decisions when discussing changes in the IT delivery process. A measurement framework implemented and used throughout the whole scaled agile IT portfolio helps stakeholders collect comparable data from different projects, achieve results that can be compared, and prioritize projects based on the same evaluation criteria. More specifically, results obtained through standard measurements are more transparent to different (cross-functional, cross-hierarchical) stakeholders. Less effort is needed to communicate results and make decisions if stakeholders discuss based on a common understanding of what is measured, how, by whom and why. In addition to more efficient communication and decision making, effectiveness of scaled agile projects and portfolios is achieved by taking measurements in an agile way, along the whole development lifecycle, adapting projects based on results obtained from measurement.

1.5.2 Sub-research questions

The central research question leads to five sub-questions presented in this sub-section. They are based on the design-oriented research of a project as presented in (Hevner, 2007). The flow of these sub-questions touches relevant areas in answering the main question and helps resolve the problem at stake.

The first sub-question aims to gather information on the new delivery model, scaled agile, in general, thus defining the application domain.

RQ1: What are the implications of adopting agile practices on IT project development and measurement, in large companies?

- ***What is agile and the Scaled Agile Framework?***
- ***What does it entail for an organization to adopt this way of working?***

This chapter looks at the main aspects of scaled agile methodology and what this change entails for an organization. It is expected that change management and measurement of success pose an important challenge in organizations taking the transition path to agility

The second and third sub-questions take the research a step further by looking into the current measurement frameworks used in IT project development, in general and in (scaled) agile projects respectively. These two sub-questions define the state-of-the art in the domain of the research as well as the grounding theory. To ensure that our solution contributes to research and is not a routine design based on well-defined knowledge and processes, we address research sub-questions RQ2 and RQ3.

RQ2: What measurement practices, metrics and measures are used to assess agile and traditional IT projects?

We focus on what indicators and measurement processes are proposed in research and used by practitioners. This step is necessary for two reasons - understanding how IT measurements are done as a foundation for building new knowledge when our scope towards effectiveness and value measurements of Scaled Agile IT projects, and avoiding any overlaps of work that has already been done, or frameworks that were intended for other situations but may apply in the case of this study.

At this point, we should have reached a consistent understanding of how IT projects are measured, and what metrics are specifically used for scaled agile project measurement of effectiveness and value. Building on this knowledge, we can proceed to the next step for answering RQ3 which incorporates novel measurements and methods from interviews in combination with existing methods identified in literature and practice, to target the specific research problem. In this sense, we turn towards IT and business practitioners from Shell to find answers about the actual and desired situation of measuring IT projects, within the scaled agile framework which is currently implemented in Shell's IT organization.

RQ3: How does the value measurement framework look like?

- ***How useful are the measures and metrics underlined in the framework for project measurement?***
- ***What is the context of application and measurement practices in the enterprise environment?***

The third research sub-question provides the prototype framework by specifying what indicators are to be measured, what are the methods of measurement and what are the prerequisites to obtain and make sense of these measurements (what governance structures need to be in place for responsibility of deciding on sources, collecting and evaluating data, how are results stored and reported, etc.). This is done through data collection and analysis of semi-structured interviews with Shell agile project and program stakeholders. Afterwards, we proceed to evaluate the measurement framework and process to design it.

RQ4: How does the measurement framework fare against practical and academic evaluation?

This final research question is meant to understand what the value of the deliverable is from a scientific and practical point of view. Because the deliverable is not assessed in a live, longitudinal case study, we evaluate it by reflecting on each step done in the research against scientific quality criteria, and assessing the deliverable against the challenges it might face in practice.

1.6 Research Approach

In this section, we introduce our research strategy including the motivation for our choice, followed by a presentation of the research methods used as tools for answering each of the research sub questions. Again, we motivate our choice for the research methods with respect to the sub questions.

The research our will conduct is strongly oriented towards producing an artifact that intends to support the solving of a practical problem and improving the situation (or domain) in which the artifact is planned to be used (in the context of our research, this being the domain presented in the Situation section at the beginning of this paper). Based on this core premise, our have chosen the design science research approach because this class of research approach supports a structured creation of this type of deliverable, as well as evaluating its applicability and value in practice. Moreover, the orientation of design science towards IT/IS research as well as its rigorous, iterative nature to solve a problem fits with our current subject and goals. From the existing design science research literature, we utilize the three-cycle view of Alan Hevner (Hevner, 2007). We motivate this choice further as our pass through each of the three cycles, describing complementing research methods used and research sub questions answered.

The first step is the **Relevance cycle** and deals with identifying the domain of application and practical context of the problem. The goals of this first step are to produce the input requirements for dealing with the problem identified and output constraints (or acceptance criteria) for evaluating the results that will be found. We address the domain step by seeking for the answer of our first research sub-question. Straightaway, the answer should be related to challenges companies face in the practical environment, a basis of requirements for the measurement framework to fulfill and thus mitigate the challenges faced. Research methods used for answering RQ1 mostly rely on desk research of scientific articles on the challenges and practices of IT project delivery (especially when agile methodologies are involved) and practitioner literature and reports on the application of scaled agile frameworks in large organizations, and the relationship between insourcing, IT-business alignment, and agile work methodologies.

The second step is described by the **Rigor Cycle,** which should draw upon the knowledge identified in the Relevance cycle and add novel findings on top of existing knowledge. This step has a scoping role of finding the right knowledge, refining it, and adding to it, as a foundation for generating an innovative and novel deliverable in the final cycle stage.

We address the second research sub question and thoroughly research what are the existing measurement frameworks used for assessing effectiveness and value of IT project development. The research methods used for answering the second research sub-question is based on desk research consisting of literature review. We will search through literature in scientific articles from the IT measurement domain (with focus on effectiveness and value measurements) using the Google Scholar search engine and the TU Delft repository. Due to a significant practical orientation of IT measurement, our will also search through practitioner white papers, case studies or reports that relate to practical implementations and observations of IT measurement frameworks and processes.

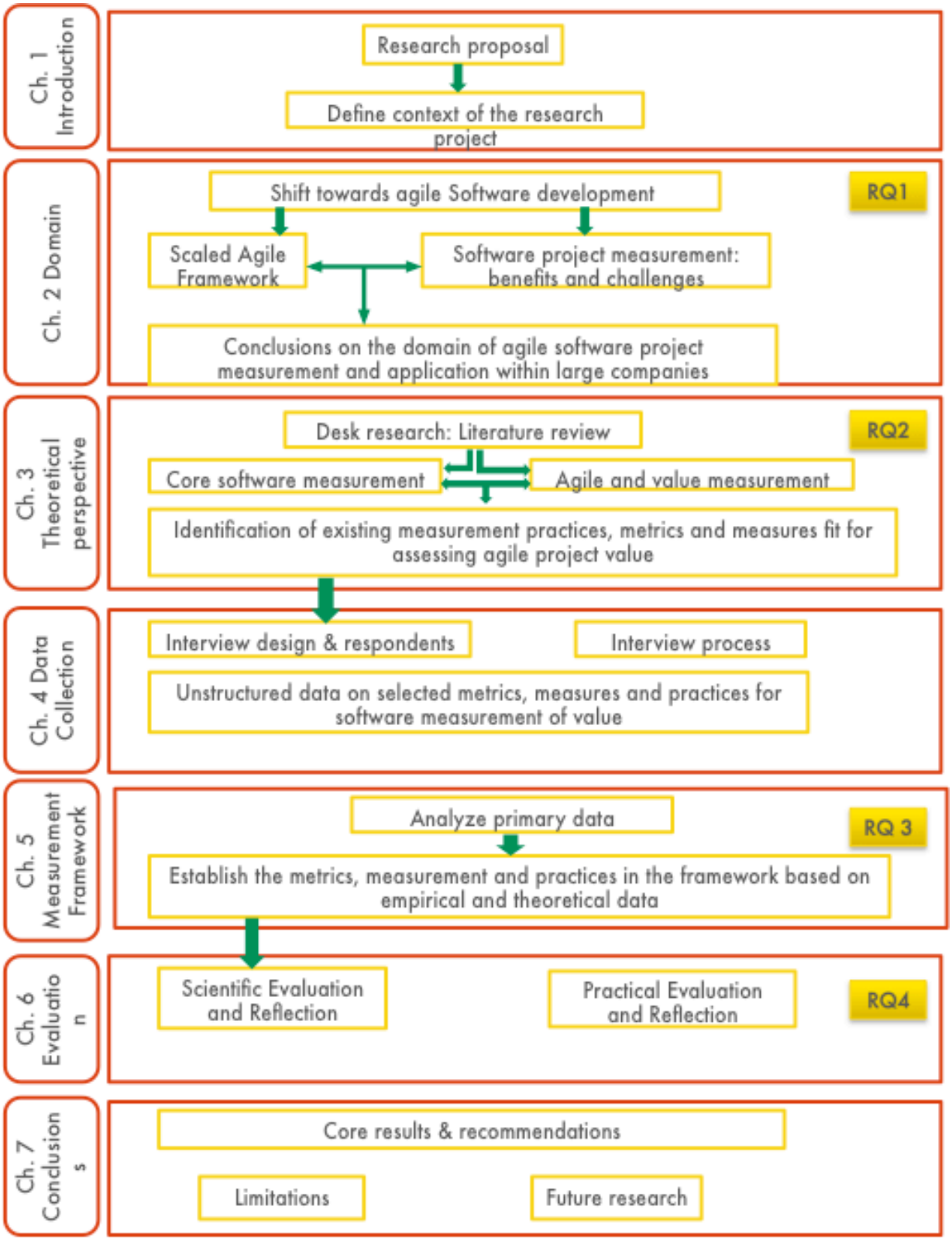
The third research sub question is also answered within the rigor cycle and relies more on additions to the knowledge base by exploring the experience and expertise of practitioners combined with findings from literature. An additional literature review of this specific topic will be required to search for any evidence of research done in measurement of scaled agile (although our have not yet found relevant scientific articles in this specific direction, except for some practitioner white papers and expert website articles).

Our intent to compare our literature findings with the situation in practice (tools and measurement frameworks that exist in literature versus the ones used in practice), and inquire what new metrics should complement existing ones in order to mitigate the lack of knowledge about effectiveness and value of scaled agile projects. This will be achieved through several interviews with Shell IT decision makers such as IT architects, project/portfolio managers, and agile project owners. The interviews will be semi-structured, containing a mix of open and closed questions to both explore and focus issues on value measurements done and those that are lacking. We choose interviews because of different perspectives needed to be applied for different types of stakeholders. The interview questions and protocol (which our will design and follow during interviews) will be adapted based on the roles of these stakeholders and their position with respect to IT measurement (do they measure, do they decide based on measurements, do they decide what measurements are needed, etc.) The answers will be recorded in writing and on tape recordings.

The final step of the cycle is prototype and evaluation. The purpose of this step is to repeatedly evaluate and reshape the deliverable until it reaches an acceptable level (judging by the requirements and constraints defined in the relevance cycle step. This cycle draws from both relevance and rigor findings, utilizing constraints and requirements to shape novel findings on the foundation of existing knowledge. The orientation towards practical applicability fits the goals of our research for providing a deliverable that can be used to measure value of scaled IT agile projects. This cycle holds the answer to the last two research questions that seek to delineate the form and contents of the measurement framework and evaluate its value and applicability in a practical environment. The research questions may be cycled through several times until our reach a satisfactory framework deliverable, based on evaluation.

The dominating research method for this step is data analysis from the previously performed desk research and interviews. Based on previous findings about measurement tools found in literature, used in practice, and complemented by novel measures and methods we extrapolate from interview responses, our can fit together the framework. The answer to the final research sub question is obtained through evaluation of the prototype by reflecting against scientific and practical quality criteria and requirements. Further evaluation could be continued when there is a case study consisting of an applied agile software development context, where a satisfactory design of the framework is achieved it fits project-specific requirements. The research methods to achieve this evaluation are cognitive and reflective, in the absence of a case study or chance to do a further interview round.

1.7 Research Flow Diagram



Chapter 2 DOMAIN DESCRIPTION

2.1 Introduction

The first chapter in this thesis exposed that the IT landscape of companies today is challenged by a struggle between outsourcing and insourcing IT capabilities and functions. We showed that IT insourcing is often desirable in companies that require higher agility, tighter IT-business alignment, and cost-cutting. To achieve these goals in-house, companies should imitate software providers' processes of development: agile methodologies. However, apart from agile adoption requiring certain organizational, culture-wise, and technical contexts, another significant challenge is that the benefits of agile software projects are difficult to measure, especially at the higher, IT portfolio level. This chapter introduces the context of agile software development and the significance of measurement, as a factor that helps evaluate and steer project value and performance. The style of this chapter is descriptive and treats the first research question:

> ***RQ1: What are the implications of adopting agile practices on IT project development and measurement, in large companies?***

To answer this question, we expose background information on software development and how agile practices emerged. This is necessary in order to understand the motivations, benefits and challenges of utilizing agile project development in companies. We emphasize on agile development in large companies, based on the scaled agile framework. Afterwards, IT project measurement is introduced as an essential but underdeveloped field in business. We explain the motivations and challenges associated to software measurement in general, and to agile project measurement in particular. Finally, the potential solution -- building a value measurement framework -- is motivated, based on the challenges exposed in the context, as solving the main research problem of this thesis: the difficulty of IT project and portfolio managers to assess the value of agile projects.

For answering this research question, desk research is carried out consisting of a literature review on the topics mentioned earlier. Articles in the area of IT sourcing, IT-business alignment, agile and waterfall software development and measurement were found to be plentiful. However, the depth and number of relevant articles in the area of agile software measurement, especially when related to value measurement, is thin. This step of the literature review is preliminary, followed by the core literature review that establishes the theoretical assumptions and concepts on value measurement in Chapter 3.

> - ***What does it entail for an organization to adopt this way of working?***

2.2 Agile software development process models

Software engineering is a discipline that is arguably over 40 years old, which has grown immensely, with a significant portion of society's functions now being supported by software systems (Sommerville, 2009). However, some of the problems of software development remain the same today as they were when the name "software engineering" was first coined in 1969 at a NATO conference -- missed deadlines for software delivery, functionality that does not match user needs, costs exceeding budgets, and reliability of software (Sommerville, 2009). These problems arise because software development is much more than a programmer writing code behind a PC screen. Software technology (the devices, platforms and skills needed to for programming and operation of applications) is integrated with software economics, human relations, and the specific context, or demand, for which a software application is developed (Boehm, 1989). These areas are integrated together under the discipline of software project management, which requires knowledge of the entire software development lifecycle from defining goals and vision to releasing and maintaining software applications (Stellman, 2005). The sequence of activities conducted to produce software applications within a software development lifecycle are orchestrated by project managers and related stakeholders by following a software process model (Sommerville, 2009). This subchapter briefly presents how and why agile process models emerged, and how they were adapted for large organizations into the scaled agile framework. Having this context information is important for understanding why agile is becoming predominant (due to its better fit with the fast-changing business, technology and market) landscape of today, and consequently, why more effort should be invested in finding appropriate measurement practices for agile.

2.2.1 Agile vs. Waterfall

Software processes models reflect different approaches to software development and should be chosen depending on the context of the company, project or team that develops the software. Plan-driven software process models such as waterfall have for a long time dominated the software development landscape, because they emphasize on well-defined process phases that follow sequentially, each starting based on the results of the previous, following a pre-determined plan (Sommerville, 2009). This model fits past software development projects because clients had a more stable set of requirements, due to less dynamicity and competition of businesses. Plan-driven approaches are still useful for some types of software, such as safety-critical control systems, where thorough analysis is required. However, in the quickly moving business environments at the beginning of the late 90s', agility and flexibility became increasingly decisive for companies to stay competitive. This means that clients demand software applications to be developed faster, and with increased flexibility for changing requirements. On the software development side, these client demands translate into adopting processes that can meet them, making plan-driven approaches obsolete in these cases (Sommerville, 2009).

Agile software development is most often defined based on the "Agile Manifesto" (Fowler, 2001) principles that value: "individuals and interactions over processes and tools", "working software over comprehensive documentation", "customer collaboration over contract negotiation", and "responding to change over following a plan". These principles are detailed and operationalized in other articles that

specify agile methods (SCRUM, Extreme Programming - XP, Feature-Driven Development, etc.). In short, agile methodologies work by a software project team incrementally adding features and functionality based on (often changing) client requirements (Sommerville, 2012). Evaluation and the necessary modifications are made at each increment (called sprint), until the client is satisfied with the results. This type of approach is close to how people generally solve problems, i.e. rather than working out a complete problem solution in advance (as with plan-driven methods), we take a series of steps on which we backtrack when realizing that a better action may be taken. The benefits of agile methodologies when compared to waterfall, for both client and software development project team, are reflected in Table 2.1.

Agile vs Waterfall Benefits	Description
Better value software delivered	Lower costs of redoing low-value functions because they are identified earlier in development.
Flexible Change in Requirements	Reduced cost and effort of managing client requirement change due to fewer amounts of analysis and documentation that needs to be redone.
Smoother Communication and Feedback from Client	Customers can comment and give feedback on demonstrations of how features function after each increment, which is easier than judging progress based on software design documents.
Quick delivery and deployment	Functional software is delivered even if not all functionality has been included, so that they can gain value from it, and test if it meets expectations from an early stage.

TABLE 2.1 AGILE BENEFITS OVER WATERFALL (BOEHM, 2005; SOMMERVILLE, 2012)

Apart from benefits, agile software development comes with some challenges. The motivations of agile adoption are based on a combination of perceived benefits and challenges (Vijayasarathy, 2012). Moreover, agile adoption strongly depends on the support and approval of key stakeholders in the adopting company, and on the possibility of training for the transition. Another essential finding is that the experience of developers and size of company negatively influence the motivation for agile adoption (Laanti, 2011; Vijayasarathy, 2012). This is because of entrenchment at both individual level (developers, project managers, other stakeholders) and at company level (Program and C-level management) who either are not motivated or directly resist change to agile. Moreover, the principles of fast changing and thinly documented work methodology of agile is in contradiction with the highly formalized and well-established enterprise structure and processes of large companies. We note that these motivations are far from being purely related to software development process improvement. Thus, the wider context of the agile adopting firm must be studied in order to understand not only if agile could, but also if it should be implemented. Another point to consider is that agile principles should not be applied exclusively, but rather, a balance should be achieved between plan-driven and agile methods to fit their own company context (Boehm, 2005).

The following subsection presents the prominent challenges that companies face when planning to adopt agile methodologies for software development. This is essential for the thesis context, as a key research assumption is that the scarcity of methods to estimate agile project value deters the benefits and motivations of agile adoption, as exposed in the problem statement.

2.2.2 Agile challenges

As exposed earlier, motivations for agile adoption must exist from multiple areas involved in a software development project. Several challenge areas that apply to switching from traditional to agile methods should be treated for successful migration, as concluded by (Nerur, 2005) in Figure 2.1.

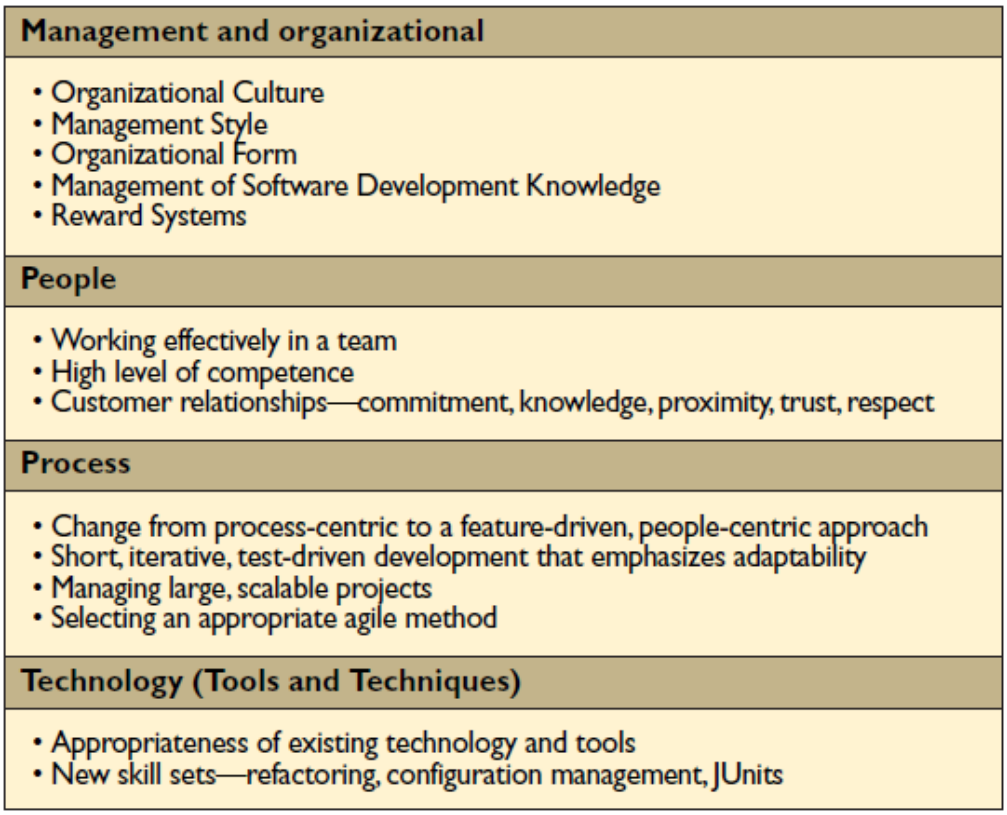


FIGURE 2.1 KEY ISSUES IN MIGRATING TO AGILE (NERUR, 2005)

As can be seen in Figure 2.1, the decades-long effort of companies to create optimized, repeatable and stable processes is being undone by agile methodologies, requiring severe changes in many areas. For example, a change essential for this thesis is in process/project management, which traditionally were compliance-driven and measurement-based, aimed at providing assurance. Agile requires speculation and planning that allows uncertainty, making measurement of benefits and assurance difficult to include. Moreover, the project and process management style must shift from command-and-control to leadership-and-collaboration, with managers becoming facilitators of collaboration between development, operations, testing, clients and product demo teams (Nerur, 2005). Alternatively, all these parties that should collaborate must be able and willing to work together, based on the channels facilitated by managers. This requires significant efforts for aligning technical and business stakeholders,

to work together so that software is not only quickly released and changed, but with increased usefulness for customers.

2.2.3 Scaling Agile for large organizations

As seen in the previous subsection, agile transformations come with significant challenges that extend above just the software development process. The transformation requires changes and support at higher management levels, and across different departments and functions. This broad extent to which change is required scales up with the size of a company and the size of software development projects (Boehm, 2005). As this thesis focuses on corporate level agile transformation and measurement, we briefly explain how agile scales with company size, and what are the incentives and barriers at this level.

At large company level, software development faces a paradoxical issue. On the one hand, large companies rely on standard, well defined processes for controlling business and support activities. This includes software engineering processes and systems that are structured and refined over the years, mostly based on customized plan-driven software development processes. On the other hand, large companies are in dire need of keeping up with quickly changing technological and market needs, so that they be competitive and at the same time reduce the cost of development activities due to shorter development lifecycles. Agile methodologies can arguably solve the latter issue, but only if the former consideration can be surpassed. Thus, the core challenge of agile adoption in large companies is implementing agile methodologies and at the same time merging them with incumbent control and structure (Boehm, 2005). As observed in several annual workshops on agile adoption in large companies (Boehm, 2005), companies successfully managed to implement agile/hybrid methodologies on small, non business-critical pilot projects, but failed to scale these projects and processes in most cases. Even in full-scale cases of agile implementations in enterprises (such as Yahoo!), teams reportedly worked with agile in various proportions, from fully Agile to “mini-waterfalls” coined as Agile (Benefield, 2008).

In recent years research has been conducted and methodologies developed on scaling agile to work at enterprise level. Techniques such as disciplined agile delivery (Ambler, 2012) or the scaled agile framework, also known as SAFe (Leffingwell, 2010) provide some indications on how to maintain “discipline”, while reaping the benefits of quicker software releases and more flexible development processes. In the context of this thesis, the company we interviewed followed the scaled agile framework, and were in the process of planning how to implement agile in several pilot projects following this framework. For this reason, we consider SAFe as the example for how agile could be scaled at enterprise level, including the approach on scaling measurement and evaluation of agile projects. Briefly, the SAFe recommends how agile and lean principles should be applied above the individual project level, presenting a map of stakeholders and the relations between them for controlling an agile portfolio. The focus of SAFe is on explaining the activities of stakeholders at various hierarchical and functional levels, across several agile project teams, while maintaining top-down control that is so often met in enterprises (Leffingwell, 2010). It is important to mention that this method is far from perfect, as it has serious drawbacks in clearly specifying how the various modules in the framework work together (Elssamadisy, 2013). The core implication of SAFe usage for this thesis is that it does not specify a unified method for estimating the business and user value of developed software. This implies that enterprises are under high uncertainty in selecting the right projects.

2.2.4 Subchapter conclusions

In this subchapter, we briefly introduced software development process models, emphasizing on how agile methodologies emerged to complement and replace plan-driven approaches, to better fulfill a market demand for software that can be easily changed and quickly developed. In addition, we presented some of the core challenges of agile adoption, which requires change across many functional and hierarchical levels across the organization. We argued that agile adoption at both small and large (enterprise) scale is challenged by the lack of acknowledged methods for identifying the value of agile software projects. Without such an evaluation, there is little evidence as input for project and portfolio level decision making for comparing different projects, and understanding their value for the business and end-users. As mentioned in the research objective, our aim is to explore how these issues can be mitigated through software project measurement, focused on measuring the value of agile projects. The following subchapter introduces the field of software project measurement, explaining its benefits, motivations and challenges, scoping towards agile and value measurement.

2.3 Software project measurement

Software measurement is a subfield of software quality management, which acts at both portfolio and project levels, purposed to establish and guide the application of a framework of organizational processes and standards that lead to high quality software (Sommerville, 2012). At project level, quality management involves the definition of a quality plan which sets out the quality goals. Software measurement implies assessing the extent to which quality goals are met, by tracking and interpreting a set of quantitative or qualitative values that characterize certain attributes of software components or processes (Sommerville, 2012). Independent of what the quality goals of software are, measurement is done differently depending on the software development model applied. Agile and plan-driven approaches impose different evaluation methods; utilizing plan-driven measurement approaches to agile proves to be not only ineffective, but counter-productive (Morasca, 2001; Hartmann, 2006). However, agile measurement is surprisingly underdeveloped in both research and practitioner literature, especially in the particular field of value measurement of software. This subchapter intends to describe why this is so, and ends by warranting further research into this area.

2.3.1 Benefits and motivations of measurement

Measurement in engineering, and particularly in software development, supports production planning, monitoring, decision making and maintenance of a product that is utilized by its target users (Morasca, 2001). Effective measurement helps software development project teams and their managers understand and predict both what resources are required for upcoming projects, and how to best use these resources to produce software that is useful for clients. Because measurement is usually done by technical people involved in development, and its results communicated to partly technical project managers and product owners, most research done up until now is focused on the benefits and challenges of measurement from the perspective of development teams.

However, the past decades have brought IT more or less on the same level with business, because of its crucial support, and in some cases driving, roles of keeping a company competitive, and making it agile with respect to turbulent environments (Overby, 2006).

This new relationship between business and IT requires much more collaboration to align the two areas, which is reflected in software development by the emergence of agile methods, and consequently, by the involvement of a wide range of stakeholders in the software development projects. The implications of these arguments for software measurement and this thesis is that measurement must move together with what it measures, and who is using the results for measurement. In other words, measurement processes should fit agile software development approaches, and the goals and results of measurement should be oriented equally towards technical stakeholders interested in efficiency (smooth and efficient development processes), and business stakeholders interested in value (high profits or other kind of benefits for the business), both working together to fulfil client demands (high usability and ease of use) (Kulas, 2012).

2.3.2 Challenges in measurement

Despite the obvious benefits and motivations for adopting software measurement programmes, measurement in applied context is rarely given the importance deserved. Some of the main reasons are that benefits of measurement are not evident, metrics programmes are costly and difficult to maintain and standardize, and there are not enough supporting frameworks to suggest how measurement can be done, especially with agile, and value-oriented measurement goals. It is often the case that software development teams measure characteristics that are not necessarily meaningful for decision-making. Such measurement is done from inertia (previous projects), or because teams follow measures that are specified in frameworks without experimenting if they are useful for the particular software development case (Morasca, 2001).

The discrepancy between having to adapt measures and approaches for each software project and the need for standardized measurement that can be followed is one of the reasons for difficulty to introduce measurement programs (Kulas, 2012). Even within the same company, there is too much process variability to use measurements in a meaningful way. This is presented as the biggest issue with software measurement, which is worsened by the scarcity of empirical research on systematic software measurement in industry, to support measurement frameworks and metrics. Another reason for this lack of empirical evidence is the reluctance to commit resources to introduce measurements, because this adds overhead to processes. Thus, despite the importance of software measurement for software development being recognized, the overhead created is seen as unjustified - business stakeholders often consider that measurement does not provide significant support for managerial decision-making with respect to software projects and their lifecycle (Kulas, 2012). The high perceived costs and low return of metric programs are most frequently characteristic to the initial phase of implementation, because there is little to no historical data to compare against. Therefore, high costs of metric programmes, lack of standardization and inappropriate use of metrics or decision making are contributing factors to this low utilization and immaturity of the field (Kaner & Bond, 2004).

Apart from these issues that apply to software measurement in general, there are the additional issues of agile software measurement and the practices. First and foremost, there is the difference between classical measurement programs and agile measurement. In general, software measurement is largely dominated by code-based metrics (metrics that focus on software application code - quality, efficiency, size, etc) for plan-driven development or adapted for agile (Sommerville, 2010). As mentioned earlier, due to different processes and practices in agile, the traditional research focused on code-based metrics and plan-driven development processes does not apply on Agile. In fact, using evaluation and monitoring methods and metrics that are plan-driven for Agile software development may determine counter-productive effects, that prevent transformation to agile and realizing increased software product value (Hartmann, 2006; Sommerville, 2010). Most authors are in accord with this argument, which advises against using measurement processes and standards as a blueprint for agile development (Kulas, 2012). Due to the focus of agile on value-delivery (feature-driven processes) and collaborative work between various stakeholders (leading & facilitating rather than managing), measurement practices need to be reoriented towards assessing value.

However, the lack of a more or less universally accepted and recognized agile measurement practice is also caused by the dynamic nature of requirements and lightweight, less strictly defined practices (Javdani, 2013). For example, traditional effort estimation in software projects, which intends to predict the effort required for implementation, is difficult to apply in agile practices. Traditional effort estimation is based on historical benchmarking of data, and estimating all tasks needed to be done, activities not present in agile development. Moreover, agile focuses on fulfilling customer needs and business value, objectives which are very difficult to quantify and include in a measurement framework (Fehlmann, 2014). It is still uncertain how this side of research applies on the relatively novel software development techniques such as agile (Sommerville, 2010), since plan-driven evaluation and monitoring methods and metrics do not apply, at least not without serious adaptations, to Agile (Hartmann, 2006).

2.3.3 Subchapter conclusions

As we have seen throughout this subchapter, literature shows that the decision for investing in software measurement programs is a difficult one, because of the potentially high risks and rewards attached. On the one hand, software measurement promises, at least in theory, many uses for ensuring that the software development process is efficient, and the deliverable is effective to meet client and business expectations. In addition, specifically for agile, measurement is needed to confirm the opinions and arguments claiming the benefits of agile development. Measurement is also required to keep track and communicate the value of agile projects, so that portfolio and project level decision-making can be data-driven. On the other hand, software measurement programs are often expensive to implement and slow to reflect true value, especially in the beginning when there is little to no historical data to benchmark against. Because measurement depends on each project context, standardization is difficult to achieve, reason for which there are very few specific measurement frameworks available in literature. Last but not least, agile project measurement adds a dimension of complexity by its dynamic, lightweight, and short lifecycle approach to software development. This makes classical, efficiency-based measurement approaches difficult to apply and interpret by decision makers, indicating that value-oriented measurements would be better suited for this type of approach.

2.4 Chapter conclusion

This chapter has described the context in which research is conducted, explaining the potential benefits and challenges of agile software development adoption, and measurement programs associated to it. The sub-question: “What does it entail for an organization to adopt this way of working” is answered.

We found that agile methodologies are more desirable than plan-driven methods, when company goals are enabling a more dynamic and adaptable software development process to meet turbulent market environments. However, agile adoption requires significant changes on multiple functional and hierarchical levels that relate to software project management and development. These changes are especially challenging at enterprise level, where there is a mismatch between the need for control and structure, and agile work practices that rely on dynamicity and lightweight processes and structures. We found that potential adopters at enterprise level are reluctant towards agile because there is little evidence to support agile benefits are achievable. There is no proven method to measure agile benefits, except for classical software measurement methodologies, which were shown to be inappropriate for measuring software developed with agile.

This situation leaves project and portfolio decision makers without a data input for decision making, particularly when seeking to find the value of software projects for the business and end users. This thesis intends to improve this situation by studying what kind of measures and measurement approaches could be used to assess the value of agile software projects. The following chapter develops the theoretical foundation, which acts as a basis for designing a measurement framework to fulfill the research objective.

Chapter 3 THEORETICAL PERSPECTIVE

3.1 Introduction

The previous chapter set the grounds for exploring how measurement of agile projects can improve decision making on agile project selection and development. To achieve this objective, this chapter analyses existing measurement practices, identifying potential methods and metrics for value measurement, with emphasis on agile. Therefore, the chapter purposes to answer the second research question.

> ***RQ2: What measurement practices, metrics and measures are used to asses agile and traditional IT projects?***

To answer this question, we conducted a literature review of scientific articles in the domain of software measurement, focusing on both plan-driven and agile development value measurement practices and metrics. We start by defining what a software measurement framework is and the elements that it should contain, as well as how it should be applied in practice. Afterwards, the most salient measurement frameworks, methodologies and metrics are analyzed, showing how they apply in practice. We then scope towards the specific goal of assessing value in agile project development, and propose a set of metrics and measurements that can achieve these goals. These findings set the requirements for the designing the deliverable of this thesis, and are refined and empirically tested through interviews, described in Chapter 4.

3.1.1 Research methods and role of literature review

This chapter is written based on the second stage of desk research, also consisting of a literature review, but different from the first review that was used to map out the research context and introduce its basic elements. The purpose here is to not only describe, but analyze and select theories that can support theoretical assumptions required for specifying how the measurement framework will be built. The challenge of this review is in sifting through the wealth of literature on software measurement, and selecting the few sources that are relevant and useful for agile and value measurement.

The existing measurement frameworks documented in research and used in practice will be reviewed to understand the drawbacks and possibilities of improvement that exist and could be mitigated in the context of this research. Performance and success measurement of IT has been a well-covered subject, albeit most methods target analysis of efficiency factors such as lines-of-code, defect density, delivery time, etc. (Zuse, 1998; Jones, 2008; De Wit, 1988). The problem of software value estimation has been recognized as difficult to assess but extremely important where business and IT work closely together (Biffl, 2006). Therefore, we explore and form a basis of research that focuses on the value of IT delivered within the current measurement practices existing in literature, especially with respect to agile project development (Hartmann, 2006; Javdani, 2013). The research methods used for writing this chapter are based on desk research, reviewing literature from scientific articles on the subject of software

measurement. The secondary data sources explored were found through scientific research search engines such as Google Scholar and Scopus. A wealth of articles was found on software measurement practices, especially ones with strong engineering orientation. To narrow the search towards the thesis topic, keywords were used in combination, among which the most common were: “software”, “measurement”, “agile”, “value”, “effectiveness”, “framework”, “methodology”, “metric”, “measure”, “performance indicator”, “scaled agile”, and others.

3.2 Analysis of software measurement

3.2.1 Overview of articles

This section shows only the selected articles in Table 2, from the totality of articles reviewed, underlining the notable contributions of each, and scoping towards the value assessment of agile and scaled agile software projects. The selection was done based on two filters to judge relevance. The first is the notoriety of measurement practices and methods, or how well known and utilized they are in software development projects. The purpose of the first consideration is to track which measurement methods are prominent, and what elements could be used in combination with novel, agile / value-driven measurement practices. The second filter is the application of measurement practices and metrics for assessing value of software projects (for both agile and plan-driven approaches). The following table presents the authors, titles and utility of articles selected for this thesis.

Author	Title	Description
Tarhan, 2014	Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process	Describes mostly efficiency metrics for development performance and product quality, and some indicators of how usable are the metrics.
Sommerville, 2010	Software Engineering 9th Edition	Introduction to the concepts of measurement in software and underlines the most commonly used indicators and metrics
Petersen, 2010	The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study	High level description of some additional metrics that are useful when moving to agile practices.
Olszewska, 2016	Quantitatively measuring a large-scale agile transformation	Underlines the importance of value measurement to observe the benefits of switching to plan-driven to agile in large scale companies.
Hartmaan, 2006	Appropriate agile measurement: Using metrics and diagnostics to	Presents a combination of financial (business value) and engineering metrics for measuring

Author	Title	Description
	deliver business value	agile software development.
Mahnic, 2007	Using stakeholder-driven process performance measurement for monitoring the performance of a Scrum-based software development process	Focuses on the measurement of the agile development process, mostly taking in account efficiency of workforce and resources used.
Fehlmann, 2014	Early Software Project Estimation the Six Sigma Way	Project estimation measures for assessing customer needs and hidden requirements through QFD (Quality Function Distribution) from Six Sigma practices, prior to development
Marciuska, 2014	Feature usage as a value indicator for decision making	Proposes usage of software metrics for decision making on features to be improved/removed
Herzog, 2013	Methods And Metrics For Measuring The Success Of Enterprise Social Software	Detailed assessment of user value metrics and indicators, combined with business value metrics for enterprise social software
Vickers, 2001	An introduction to function point analysis	Describes one of the most popular methods for software size and effort, Function Point Analysis
Dale, 1992	Software productivity metrics: who needs them	Describes metrics for IT expenditures (investments), their value and productivity.
Banker, 1994	Automating Output Size and Reuse Metrics in a Repository-Based Computer-Aided Software Engineering (CASE) Environment	Automation of metrics with computer aided software engineering, focusing on efficiency.
Zuse, 1998	A framework of Software Measurement	Introduction to software measurement motivations, benefits and challenges.

TABLE 3.1 SELECTED ARTICLES TREATING SOFTWARE MEASUREMENT

The articles generally describe metrics that are oriented towards efficiency measurements of the software development process and product in terms of resources used and their costs, compared to the perceived financial benefits. Also, there are many articles focusing on how to estimate the effort needed for development. However, there are some sources that admit the significance of measuring value, and that agile methodologies need different kinds of metrics and methods as opposed to classical plan driven approaches to software development. The following section details the findings and their relevance for this thesis.

3.2.2 Software measurement core concepts and activities

In the context-building chapter, we described the motivations for software measurement, along with challenges of measurement in agile software development. This section discusses the core activities and structures that software measurement implies, regardless of the software development processes they are applied based on Sommerville (2010). From the previous table of sources, Table 3.1, we selected Sommerville because he provides an overarching view over basic concepts of software measurement necessary at this stage, while the other articles go into more depth scoping towards specific measurement practices. The following table, Table 3.2, underlines the core software measurement definitions and terms used in software measurement, and briefly discuss their relevance for the final goal of this thesis. The drawback of this source is that it has a clear influence of plan-driven software development approaches, despite discussing general measurement practices and concepts.

Concept	Definition	Relevance for thesis
Software Measurement Goals	The reason for which software measurement is conducted, with respect to a subject or entity of interest (Fenton, 1994). Traditional goals are assessing the effectiveness or efficiency of software development process and its deliverable. New complementary goals aim for understanding value and usability of the deliverable.	The goals of measurement are compulsory for choosing the right metrics and methods to quantify them. Assessing the value of software implies a largely different approach to measurement than with traditional methods.
Software Metric or Attributes	Also known as indicators or attributes, metrics are the characteristics of the software components, systems or processes that are evaluated through software measurement. The metrics indicate the level of conformance with detailed product specifications, thus specifying if goals were attained or not.	Metrics are high level characteristics of software that are followed, for evaluating software against specific goals - such as the value of software for business, or for users.
Measure	Measures represent directly quantifiable entities that are gathered in a measurement process. They are evaluated independently or combined together, to characterize various software metrics.	Measures are the basis for software measurement process, identifying and selecting the right combination of measures is one of the main challenges for quantifying software metrics.
Software measurement activity	Deriving numeric values to quantify attributes of software components, systems or processes, and comparing these values to each other across organizational standards, to draw conclusions about software quality and effectiveness. (Sommerville, 2010).	The core activity that is conducted to output results indicating (among others) the value of software.

TABLE 3.2 CORE MEASUREMENT CONCEPTS AND TERMINOLOGY

Based on Table 3.2, we can understand the flow of software measurement, which we have represented graphically in Figure 3.1. First, the intended goals of software measurement are defined, with respect to a software deliverable or the process of developing it. Second, we specify software metrics or attributes that act as key indicators of the goal's completion. Third, we decide the measures on which data must be collected, to quantify the software metrics. Fourth, the software measurement activity is conducted to collect the quantitative (and sometimes qualitative) data that represents the measures, and analyze it by combining different measures that can give insights into metrics. Lastly, based on the results, conclusions are drawn about a software product, to be used by decision makers for assessing if the software product fits with intended performance goals, and perceived value by users and business.

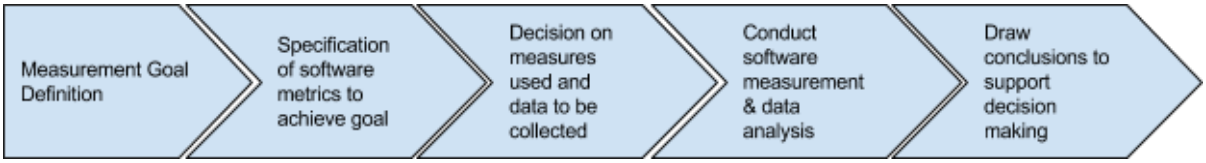


FIGURE 3.1 SOFTWARE MEASURMENT ACTIVITIES FLOW. BASED ON SOMMERVILLE, 2010

Apart from the distinction of steps and concepts described above, we underline that software measurement is different depending on the goals. The first broad distinction is between the usage of measurement for assessment and/or prediction. Measurement can be applied on finished, or underway software projects, to understand their performance and value with respect to intended targets, so that necessary adjustments can be made. This type of assessment also stands as a basis for predictive measurement, which is the other way of using measurements. Prediction is useful for the correct resource allocation and managing expectations for future projects, based on learnings of past projects (Fenton, 1994). For our research, these types are intertwined, as agile projects change so quickly that assessment and prediction are done within the same project, to redefine goals with respect to new specifications, and the resources needed to fulfill the goals.

Another distinction applies on the software metrics or attributes. This is important because it specifies what types of metrics are useful for value goals or effectiveness and performance goals. *Internal* software attributes are the most commonly used metrics - they characterize the software quality and the effort needed to develop it, from the developer's perspective. Usual internal attributes are: security, testability, complexity, modularity, and size. Common metrics that are used for measuring internal software attributes are: lines-of-code or function points to measure size of the product, number of faults in delivered software as a quality estimate, code complexity metrics (how well the code is commented, depth of conditional nesting, number of dependencies and function calls), and effort and time of development estimates (testing effort, defect repairing effort, etc.). Internal software attributes are easily quantifiable, but difficult to translate into advice for actionable decisions, especially if looking to estimate the product value from a business or user perspective. This is why they have to be linked with external software attributes. *External software attributes* are more useful for assessing product value because they relate to how the developers and users experience the software, ensuring that the software has the required properties to be usable and satisfy users. These attributes are: accuracy (fit to customer need), maintainability, usability (or ease of use and comfort), reliability, and reusability. The downside is that they are difficult to measure and quantify, and must be related to internal software

attributes. These relationships are problematic to find, due to their dependence on the context of development process, technology used and type of product developed.

3.2.3 Three levels of measurement: efficiency, effectiveness, value

As pointed out earlier, software measurement is done by establishing goals, and evaluating external and internal software attributes, measuring them against goal completion. Different goals can be set depending on the project functional or business area. Software developers, team leaders, product owners, project managers and end users, each have different areas of interest and expectations from a software project. To distinguish these goals and understand corresponding measurement activities, Dale (Dale, 1992) describes three levels or goals of a software projects, as seen in Table 3.3.

Level / Goal	Description	Measurement activities of interest
Project (Efficiency)	Level focused on efficiency of planning and controlling resources (technical) to develop software, so that certain targets and constraints can be respected.	Measures used for measurement at project level are mostly oriented inwards, to the characteristics of the code and technical specifications of the software - size, complexity, etc. Main stakeholders for this measurement category are technically oriented software development team leaders and engineers.
Delivery (Effectiveness)	Level that is focused on a merged set of characteristics from both business and project levels, concerned with the effectiveness of the whole system of delivery and the deliverable itself.	The effectiveness of the project is measured through delivery of useful functions to users, as per the specifications, the ratio between user cost and user benefit delivered, and the productivity of the team and development processes in place. Stakeholders of this category are represented by product owners that bridge between business and project stakeholders.
Business (Value)	At business level, the core interest resides in the perceived value that the software can generate for the business through its direct usage (internal), or sales to end-customers.	Measurement focuses on assessing the direct or indirect benefits (usually financial) gained from utilizing or commercializing a software product. External software attributes and financial measures are used to assess the business value of software, because they show how users experience the software. This level's stakeholders are business executives, solution sponsors, or clients interested in the financial and organizational benefits that the software may provide.

TABLE 3.3 GOALS OF SOFTWARE MEASUREMENT (DALE, 1992)

This classification of measurement by Dale (Dale, 1992) is useful for tying the stakeholders that utilize the measurement results for decision-making, with the actual types of measurement and metrics

needed to obtain results. This description is also a key assumption that motivates why value measurements become more significant, due to the increasing collaboration between business and technical stakeholders, and users. As specified in the problem statement, this thesis views the software measurement problem of agile projects from the perspectives of software development team leaders and engineers, portfolio and project managers, product owners and business executives or clients.

However, Dale (1992) does not tie the three areas and their measurement to modern software development methodologies, such as Agile, and neither is project or company size considered as a factor for determining measurement practices, gaps which our thesis intends to fill. More specifically, on large company level, measurement should focus on value of projects within the scaled agile framework. The argument of using value considerations as driving the measurement process fits perfectly for the scaled agile case. On the one hand, at project level where measurement was classically made and used for decision making, actors can understand and communicate based on technical measurements and their results because fewer functions and hierarchies are involved. On the other hand, measurement at portfolio level demands abstraction of information for decision making, making value-based metrics more relatable to the entire spectrum of functions. Thus, the following subchapters discuss measurements that are more suitable for measuring value, with agile methodologies, at enterprise scale in particular, culminating in an analysis of existing methods and metrics that will be empirically tested and formed into a measurement framework in Chapter 4.

3.3 Agile and Value Measurement

We have observed that literature mostly treat efficiency and effectiveness of software development products and processes as the most commonly used and useful measurement practices. However, we argue that value-based measurement of software becomes more prominent due to the shift towards using agile software methodologies, which focus on delivering value to customers and the business (Hartmann, 2006). As explained in the problem statement, this thesis fills this gap and intends to create a framework for value measurement of agile projects, in large companies. This subchapter details some characteristics of, and motivations for, using value measurement based on literature and then describes measurement of agile projects.

3.3.1 Value Measurement

The objectives of value measurement are to assess the success of software projects and processes from user and business perspectives (Herzog, 2013; Dale, 1992; Favaro, 2003). Business value based metrics are strongly related to financial measures and organizational benefits that software brings, while user value based metrics assess satisfaction and benefits that users experience during utilization (Herzog, 2013). While most studies show that the value domain is critical to software project success, quantifying the benefits of software and accurately identifying its beneficiaries are difficulties that block value-based software measurement from reaching a mature state (Biffl, 2006).

In a case study on measuring enterprise social software (Herzog, 2013), the methods and metrics described indicate a tendency towards value-based software measurements. The success of an information system is tied to its end-user satisfaction and usefulness. He splits software value into two

categories depending on the stakeholder and their interest with the software - organizational (or business) and user value. The author recognizes the gap in research between these two types of values, and proposes that they should be linked together to understand software value as a whole. On the one hand, user value is measured by a set of metrics that focus on usage and the benefits from end-users' points of view. These kinds of measurements are difficult in pre-development phases, and can be done only based on use cases and requirements testing with users, and not on the actual software deliverable. They require some form of preliminary functioning prototype to be tested against the users. On the other hand, business value methods and metrics demonstrate the economic value the software brings to the organization or other business stakeholders. As opposed to most business value metrics, the metrics proposed by Herzog cover both financial and non-financial indicators.

The relevance of software usage to measure value is confirmed by another study (Marciuska, 2014) which found correlation between feature usage and customer perceived value. This article proposes measurement of value to be done by comparing the intended usage that developers envision, and actual feature usage of end-users. Combining the two usage measurements can show results about business and user value, and also on the effectiveness of development teams to deliver needed features. However, these metrics should be combined with financial measurements, to assess not only the usage but the costs and potential returns of software to the business. Apart from Herzog, 2013, the usage of such metrics has been for a long time recognized, even before agile methodologies appeared (Dale, 1992). Business value depends on the cost/benefit calculations of various factors such as maintenance, IT assets, IT research investments, operating expenses, etc. Both Dale, 1992 and Herzog, 2013 claim that these metrics should be combined with technical metrics within the same measurement suite, to understand project value. These metrics and others more specific to agile development are discussed in the following subsections as components for the theoretical foundation formed in this chapter.

3.3.2 Measurement of methodology value versus project value

Before diving into the metrics selected as theoretical foundation for the agile measurement framework, we differentiate two types of agile measurement based on the object that is under assessment. This distinction is between assessing the agile methodology and assessing the product (software). This confusion between measuring the development approach and the software deliverable is often made in software measurement research (Hartmann, 2006). For this thesis, the distinction is important because this research focuses on the assessment of the project through agile development, and not on the agile methodology itself.

While our thesis focuses on agile project value rather than the value of the methodology, it is useful to understand the latter. While agile clearly grows more and more in popularity and adoption in organizations, there is still not enough empirical evidence to support its value over traditional methods (Olszewska, 2016). Empirical evidence is required to assess the impact of agile transformation, because transformation effects such as financial costs, work disruption and development quality may be detrimental. As reported by Olszewska, 2016, most evidence is of qualitative nature and a quantitative metrics model for agile transformation impact is required. This study confirmed the value of agile methodologies through a case study of agile transformation using quantitative metrics. The results proved significant agile benefits of agile methodologies over plan-driven methods in six out of eight

measures assessed. Another study used a combination of quantitative metrics, in this case, researching the development performance and product quality of Agile processes as opposed to value (Tarhan, 2014). The results supported the superiority in terms of quality of performance of agile processes and products.

Another category of studies that relates to agile methodology value measurement is agile product value measurement. Our research focuses on this side of measurement, which is more difficult to quantify, because while agile methodologies follow some prescribed practices and rules, software products are completely dependent on client requirements, rarely being the same. Thus, a standard way in which to measure the value of agile software products is extremely elusive, because there are so many variables and specific cases which cannot be predicted and included in a checklist of metrics that must be gathered (Hartmann, 2006). The article of Hartmann et. al., proposes a set of heuristics for agile measurement, and recommends that each project should come with its personalized design of a measurement framework. The definition of one core metric is recommended, related to the economics of the software investment. It should measure value holistically from the perspective of various stakeholders (development, research, marketing, sales). This method encourages collaboration within the enterprise, combining measurement of financial, engineering and user-defined value metrics to capture multiple dimensions of project value. However, the issue with this method is in its vagueness and difficulty to apply in practice. First of all, there are little cues on how stakeholders from different fields should collaborate. Secondly, an increase in process transparency due to measurement may be unwanted by some roles, who do not wish to be exposed (Hartmann, 2006). Thus, measurement of project value is strongly dependent on the people that will be both conducting, and the subjects of the measurement. This determines lots of variability in how a measurement framework can be applied, which in turn, can be a cause for the lack of more specific agile project value measurement frameworks.

3.3.3 Agile software measurement

Agile measurement is underdeveloped in practice and theory because, as we saw earlier, a standard measurement plan cannot grasp the full range of diverse agile project implementations, and the quick and volatile development lifecycle. Also, as found in literature, most plan-driven measurement approaches have failed to be successfully applied for agile, because a high number of differences between the two models. The motivations and challenges for measurement in agile software development have been described in the context building chapter. We now explain the metrics and methodologies used to assess agile software products, focusing on their value for business and users.

Practices and metrics can be gathered from a handful of scientific articles, after which we observed repeatability of the metrics and measures, which were applied only in different contexts. When intending to measure value of projects, there is no separate method that focuses exclusively on value-related metrics and indicators. This side is usually included in agile software measurement as a whole, due to the difficulty to use only value or only efficiency measurements for assessing agile projects (Favaro, 2003). Thus, we discuss the literature findings of the main practices of agile measurement, researched and detail these metrics by logically combining the findings from several papers (Hartman, 2006; Petersen, 2010; Sommerville, 2010; Herzog, 2013; Javdani, 2013; Marciuska, 2014; Fehlmann, 2014; Olszewska, 2016). A final metric list is presented, exposing selected metrics for measuring value in

the case of agile project development. The full list of metrics out of which the selection was made can be found in the annex.

The process of selection was done by categorizing metrics or KPIs and their subsequent measures into the three broad categories described earlier in Table 3.3, which are effectiveness, efficiency and value. As we have argued based on literature findings, efficiency and effectiveness are focused in plan-driven measurement methodologies, with value being seldom treated. As agile methodologies promote business and user-valuable software delivery, from the three types of measurement goals presented in Section 3.2.3, we excluded efficiency and effectiveness metrics, and focused on value. Because the concept of value is still to general, we break it down into three components, which reflect for whom value is created – the business (the software owner) or the end-users (the software user). These components are: satisfaction, uptake and usage. This categorization is not referenced in previous articles and was decided upon (in collaboration with Erwin Mul, from Shell) for simplifying the search for metrics and measures. Table 3.4 shows what each of the three components means and what type of value they reflect. The type of value can contribute to users, business, or both.

Value components	Definition	Type of value
Satisfaction	The degree to which end-users are satisfied with the software solution provided in terms of ease of use and fit with user requirements	Both User and Business value. Satisfaction is a predictor for long-term usage of software and efficiency of users when using the software. Satisfied users get the most out of software, which in turn benefits the business.
Uptake	The adoption coverage in terms of breadth of users, and growth of adoption. This tracks if the software is used by multiple types of users and for multiple purposes.	Both User and Business value. Uptake reflects the impact range of the software. The more diverse adopters, the greater the value for business. Also, collaboration and transparency between various groups based on the software indicates value for users.
Usage	The adoption of software regardless of function and uniqueness of users. Usage tracks how much an application is accessed and how it is used (user behavior, features most/least accessed, number of users, etc.)	Business Value. The usage reflects how much and how an application is used, which is beneficial for business decision makers for assessing if the software is used as intended, and by enough people to yield the appropriate returns.

TABLE 3.4 COMPONENTS OF THE VALUE MEASUREMENT GOAL

By using this categorization as a filter for the entire set of metrics and measures (see Annex), we selected a set that matches the descriptions of usage, uptake and satisfaction. The set serves as the core theoretical assumption, as seen in Table 3.5, and is at the basis of the agile value measurement framework. To support the assumption, and define structural specifications for the framework, we

conducted interviews to gather empirical data on how this set of metrics fits for agile project value measurement, part which is presented in the following chapter delineating data collection and analysis. The three interview respondents were selected due to their managerial roles in several projects that followed or were planned to follow agile methodologies. More information about the interview respondents, their selection and the questions they answer is provided in Subchapter 4.2

KPI / Metric / Indicator	Sub-indicator	Measure	Description	Value Type
Fulfilment of market needs (Petersen, 2010)	*None*	No. of change requests per requirement	Fewer change requests and less discarded requirements indicate that current market needs are fulfilled	Satisfaction
Feature Quality (Olszewska, 2016)	None	Number of external issue reports by users	The metric exhibits the quality of delivered features by comparing the number of user issue reports across similar projects, over specific time intervals within project development.	Satisfaction
Customer satisfaction (business and end-users) (Mahnic, 2007)	*None*	Quality of product, price adequacy, reliability in terms of time and costs, completeness of product delivered at the end of each sprint or release, flexible handling of changes in requirements, good collaboration with the development team	This collection of measures targets business and user satisfaction with a developed product. Measurements are made through surveys conducted at the end of each sprint/release of a feature, or entire software.	Satisfaction
Customer needs and hidden requirements (Fehlmann, 2014)	Voice of Consumer	Big data: Helpdesk tickets, feedback from support interventions, and topics discussed on user forums	These measurements are classified as VoC (voice of consumer) methods because they measure hidden requirements and customer needs by assessing their helpdesk requests, communication with support, topics of interests on user forums, etc.	Satisfaction
Feature usage (Marciuska, 2014)		Relative actual usage of features by end-users, intended usage by developers, actual usage threshold	These measures assess if a feature is underused or misused by end-users by comparing them to usage thresholds and intended developer usage. Usage below a certain threshold indicates if candidate or existing features should be removed or improved.	Usage
Usage Value (added value from users' point of view) (Herzog, 2013)	Content & Usage Analysis	Adjusted Ideas (number of ideas about features, functionality and software)	A high number of (user) ideas implemented reflects current, and predicts future satisfaction of users.	Satisfaction
		Intensity of collaboration (number of users interactions with each other based on software content or functionality)	Reflects value derived from collaboration and communication of various users by making use of the software.	Uptake
		Degree of cross-linking (range of users linked together)		Uptake
	Database and Log File	Log-in number and frequency	Quantitative measurement of database and logs on how much and how the software is used over time. Intensive and	Usage
		Feature number of times used, frequency, duration		Usage

KPI / Metric / Indicator	Sub-indicator	Measure	Description	Value Type
	queries	New users gain rate OR unique visitors	growing usage of software should indicate that it is valuable for both customers and business	Uptake
		Hits per time period		Usage
		Number of sessions, duration, average time per visit, frequency		Usage
	User interviews and surveys	User requirements	Qualitative assessment of how users perceive that their needs are met, and that the software is easy to use in given use cases and following the most common usage behaviours.	Satisfaction
		Usage behavior (sequence of steps, inputs, etc)		Usage
		Use case validation		Usage
		User satisfaction with tools and processes		Satisfaction
		Applicability of tools / knowledge of users about their applicability		Usage
Business Value (added value from business point of view) (Herzog, 2013)	User Interviews & Surveys	Effort of utilizing the software/tools	Assessing the effort for utilizing software and associated tools by end-users indicates user satisfaction and acceptance. Low effort means high satisfaction and efficiency of users, which creates business value.	Satisfaction
Size and effort for development (Banker, 1994; Vickers, 2001)	Reusability	Was the application developed to meet one or many user's needs?	These metrics are part of the plan-driven measurement practices, and are useful for assessing if software can be extended to multiple from functional and compliance points of view.	Uptake
	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?		Uptake

TABLE 3.5 VALUE METRICS AND MEASURES EXTRACTED FROM LITERATURE

As seen in Table 3.5, metrics tend to directly reflect end-user value, which consequently influences the perceived business value of software. Please note that the colors are purely for reading purposes and do not have any significance. We excluded direct business-value metrics on purpose, because they are mostly indicated by general financial measurements of projects, with little particularity for the case of agile (such as net present value, internal rate of return, return on investment, cash flow, etc.). The particularity of applying financial measurements for software development is to purposively combine them with engineering and user value metrics, to holistically capture the value of an entire project (Hartmann, 2006). As specified earlier, this combination can only be achieved by involving a wealth of roles and stakeholders in developing and implementing the measurement plan (Mahnic, 2007).

3.4 Chapter conclusion

This chapter has presented the theoretical assumptions of the thesis by analysis of scientific literature on software measurement, focusing on agile and value measurement of projects. The goal was to show that existing literature scarcely discusses agile measurement through value-oriented metrics and measures, and that this is much needed if agile projects are to be measured, especially at enterprise level, due to a lack of fitness of classical efficiency & effectiveness measurement methods. First, we

underlined the basics of software measurement, and delimited three broad goals of measurement – project efficiency, effectiveness and value. We found that software measurement in the context of agile and scaled agile development requires a focus on value, while plan-driven projects are traditionally measured by effectiveness and efficiency. Due to agile software development becoming dominant in the past decade, even adapting on enterprise level with scaled agile, we argued that value measurement of software should be researched to fill a gap in research and practice. Furthermore, we proposed to fill this gap by compiling a set of value-oriented metrics and measures from literature. We presented a list of the selected metrics and measures, showcasing how user and business value can be assessed by measuring their satisfaction combined with software usage and uptake. Moreover, we argued that this set of metrics should be combined with efficiency, effectiveness and financial metrics to achieve a holistic project measurement plan. This is included as a recommendation within the measurement framework application, after it is fully designed following the next chapter.

The next step of our research intends to test these assumptions, focusing on how the list of selected metrics and measures would fit for software project measurement. This is done by collecting and analyzing empirical data from relevant stakeholders from a company that undergoes an enterprise-level transformation towards scaled-agile project development. The next chapter describes the data collection process, explaining the choice of semi-structured interviews as the collection method, and reflecting on the data that can be obtained in the context of the interview respondents and case at hand.

Chapter 4 DATA COLLECTION

This chapter is organized as follows. In section 4.1 we describe the data collection method (interviews), motivate why it was chosen and describe the interview respondents' roles and protocol used. The chapter concludes in section 4.2, where we underline the core topics of data collection, afterwards going into Chapter 5 – Data Analysis and results, where the framework is presented based on three core specifications.

4.1 Data collection – Interviews

We have chosen qualitative data collection done through semi-structured interviews to explore the measurement practices and challenges within agile project development. This exploratory approach fits the still novel and shifting landscape of software measurement in agile. This assumption, formed prior to empirical data collection, was based on literature findings and discussions with subject matter experts. We have found that IT project and portfolio stakeholders are uncertain about what metrics and measurement strategies should be adopted for assessing value, and that diverse approaches exist depending on the IT project, the stakeholders involved, their skills and prior experience with measurement and agile, and the organizational structures responsible for IT project development and delivery. This multitude of factors that influence how measurement is done restricts us for now to assess metrics and measurement practices quantitatively, because of two reasons. One, there are too many variables influencing how value is measured (or not measured). As we argued in the theory chapter, agile methodologies apply in various ways, and are molded to fit a specific project, rather than restricting it (as the case with plan-driven methods). Two, we do not yet understand the interplay and causality relationships between the multitude of contexts for agile development, and the measurement approaches adopted (or failed to be adopted). For these two reasons, we argue that a semi-structured, qualitative, interview approach can create understanding into why some metrics and measures are used (or not), and how should a measurement framework should look like.

4.1.1 Interview design and respondent selection

First of all, we explain the semi-structured interview characteristics and how they fit to the data collection objectives. Semi-structured interviews are designed to follow a structure of questions that keep the topic on track, but leave way for the respondent to explain and give his opinions on the questions asked, even diverging into out-of-topic directions. On the one hand, respondents can give interesting insights into the subject, and get more engaged into a discussion rather than a one-sided talk (as would be with a structured interview). We are interested in letting respondents explain their experience with agile project measurement and value measurement also because there is no one respondent that knows to answer about all metrics and measures we have found. On the other hand, to minimize the risk of diverging from the topic too much, a structure of ordered questions is followed.

For posing questions that are both relevant for research and at the same time relatable for respondents, it is essential to consider two issues. First, the project that respondents work on is the most important factor that determines which interviewees to choose. We selected projects that were run with agile or in

transformation from waterfall to agile methodologies. Also, the project areas were selected such that their clients were internal (for Shell staff only), external (for Shell clients and partners), and both internal and external (for example, HR recruitment). The reason for this selection is to find out how this particular context factor – the target clients of a project – influences what metrics are relevant and usable. While a measurement framework applicable for a majority of cases is difficult, we intend to understand what specific sets of metrics are applicable in the current case under study. The case we are discussing about is one where the measurement framework should apply to a large number of various projects, managed by multi-disciplinary and decentralized stakeholders. We reflect on the appropriateness of interview respondents and measures selected for this case after the data analysis and results. The second issue for forming the right set of interview questions pertains to the roles of respondents. They should be able to understand interview questions and be able to give responses that are relevant, based on their experience and current roles.

Following these characteristics, and based upon what projects and people were available for interviews within Shell's IT organization, we expose the interview respondents' profiles and the target projects they worked with in Table 4.1 Interview respondents. It is important to mention that the level of confidentiality required by our respondents is high, so we provide only the functions and abstraction of project names to honor this requirement.

Respondent Role	Role description and respondent experience	Projects description
HR IT Portfolio Manager	Working together with business analysts and project managers to ensure HR IT project goals are delivered. Experience with HR Information systems, understanding of applications and the processes they support, and managing application lifecycle. Has knowledge of agile methodologies	HR Strategy data integration project within 40 different applications. Applications have both internal (Shell) and external clients (partners for HR function). Agile delivery has been recently introduced and plan-driven methods are still largely dominant.
IT Solutions Architect	IT consultant for the past 24 years, focusing on Agile project delivery in the past 4 years.	Internal Shell projects only, mostly developed with traditional plan-driven methods but shifting towards agile delivery in some cases.
IT Product Owner	Consultant on IT web development projects, business analysis, release planning and product ownership. Experience with traditional methodologies, and Agile methods in the past 4-5 years, presently with emphasis on agile transformation.	Shell websites projects that are external (client-facing). Strong customer focus of projects. Project works almost fully with agile methodologies, adapted to fit the Shell work environment

Respondent Role	Role description and respondent experience	Projects description
	Product owner on the delivery team side (rather than business side), works locally with business stakeholders and remotely with development teams distributed in several off-shore sites.	practices and context.

TABLE 4.1 INTERVIEW RESPONDENTS

As we can see in Table 4.1, the three interview respondents have leading management positions in three diverse areas of IT. Their expertize consists of a mix of business and technical skills for managing IT project and portfolio application lifecycles. All respondents have knowledge of, and are working with agile project delivery, as part of a larger scale transition from plan-driven to agile methods. Their profiles make excellent candidates for observing how measurement was done in the past (with plan-driven methods), how it is done presently, and what is lacking and could be improved. Moreover, their high-level, management positions make the respondents more interested in the value perspective of measurement, rather than efficiency (as is often seen with more technical roles within development teams). Having understood the choice of data collection method, and the proposed respondent roles, we continue to describe the interview protocol and how the interviews were conducted.

4.1.2 Interview protocol

The interview protocol and process was designed following some of the guidelines described in "Research Methods for Business" (Sekaran, 2013). The interview protocol is split into four sections that flow as follows. First, an introduction explains the study goals, the topic context, expectations on what should be obtained from interview responses, and the potential benefits the respondent could obtain out of the interview and study overall. The introduction section also goes over the core areas that should be covered in the interview, the time needed, and asks for confidentiality requirements and permission to tape record the interview. While this section is lengthy, it is meant to build rapport and credibility with the respondent. Credibility and rapport are essential especially because the topic and questions relate to a subject that is under development, and may touch work issues that are sensitive, potentially creating reluctance of interviewees to respond.

The second section of the interview asks about the professional background in IT project/product lifecycle management, and experience with agile versus traditional plan-based delivery methods. This section is for building the respondent's profile, to assess if their answers are based upon experience and knowledge that are relevant to the topic. Also, this information is valuable for understanding the perspective from which these stakeholders view agile project measurement (business, end-user, technical), and what importance they attribute to value measurement. Finally, we must find out what degree of practical experience respondents have had with agile work methodologies and transition from traditional development. This experience can reflect through working in one or several agile projects, or at least participating in projects that are to be moved, or in transition from traditional to agile work.

After these sections where context is built around the subject, and a profile is made on the interview respondents, the main body of the interview follows, with questions addressing knowledge and use of

metrics and measures. In this section, the set of metrics that were assumed to be relevant for measuring agile project value in the theoretical chapter are presented, and respondents are asked which of these metrics they recognize, and which of them are considered useful. A strong emphasis is placed on what metrics could be used in the future, as beneficial to the projects respondents are working in. In addition, respondents are encouraged to propose new measures and metrics that they consider relevant for value measurement, and asked about how they could be incorporated within a measurement framework for agile value. There are also questions about how measurement practices should be different from traditional methods, and how new measurement should be applied. Application of new measurement refers to technical and political / organizational considerations. On the one hand, it is important to find if measurements of various indicators can be made on software usage, uptake and satisfaction by using tools or embedded mechanisms in delivered software. On the other hand, the interview aims to find how measurement is included within development and delivery processes, how it could be interpreted and acted upon, and who the stakeholders responsible for each of these steps are. Thus, finding which metrics are relevant is only half of the problem, while the other half represents how these metrics can be applied so that the right people (business and product decision makers) can use measurement results.

Finally, the closing section of the interview protocol inquires on any additional information and opinions about the subject discussed, or about the interview process itself. The respondent is thanked for his participation and a follow-up confidentiality agreement is sent. The confidentiality agreement contains a table asking the level of permission for disclosing names, roles and other sensitive information, as well as a transcript of the discussion and information that is used in the thesis. With the respondent's agreement, the information is used in the thesis.

4.1.3 Interview process

Having objectively described the protocol that was to be followed for conducting the interviews, this section exposes how the interview process went, and what preliminary results were derived from the interviews. This section is a precursor of *Data analysis,* the subchapter that goes into detail with interview responses and draws empirical conclusions.

Data collection consisted of three interviews held via conference call with Shell employees. These people were found and contacted through the Shell internal network and through the recommendations of our Shell thesis supervisor. Before having the interview, each person received a short briefing, to introduce them to the subject and make the interview process smoother. Afterwards, during the interviews, discussion went more towards an unstructured type. Responses were seldom keeping the intended track of the interview protocol. This is because respondents did not have a structured image of what a measurement framework should contain and how it should be implemented, and so they diverged into answers and discussion points that were more familiar to their experience and current work. As our will explain in more detail in the data analysis section, this lack of knowledge can partly be attributed to inconsistent and non-standard use of measurement across projects and functions, and due to still incipient usage of agile methodologies on enterprise scale. Nevertheless, while the answers were less precise and clear for indicating the right measures and metrics than expected, they did reveal many other things. Respondents have been very open on how they envision running and measuring the value

of their projects. They exposed why there are no clear measurement guidelines, what challenges business and delivery teams face for measuring value and utilizing agile methodologies, and also the barriers of organizational context.

Thus, the limitations of this data collection method for agile value measurement are that answers received are not precise enough to form a solid basis for developing a framework to be directly used in practice. Furthermore, the limited number of responses also contributes to weaken the data, as it is not enough to assert that it is reliable. However, the strength lies in the explorative power due to the rapport that can be built with interview respondents. Other data collection methods could not have provided such an in-depth view on how measurement is perceived with both its challenges and benefits, in the Shell enterprise context. We argue that based on the learnings from this study, surveys could be more precisely designed to collect a larger and more meaningful set of data, that is descriptive (rather than explorative) of the measurement situation in agile project development.

4.2 Data collection conclusion

This chapter has described the methodology and execution of the data collection phase of the thesis. Qualitative data was collected through semi-structured interviews with three key managerial roles within Shell's Global IT functions. The respondents were represented by an HR IT Portfolio Manager, an IT Solutions Architect, and an IT product owner. The interviews were held in a semi-structured way in order to explore the relatively new subject of agile projects and their measurement, allowing respondents to expose their own ideas and experiences about the topic, while at the same time maintaining the focus within the scope of this thesis. We have found that the respondents were eager to discuss the subject, and openly admitted the existing gaps in measurement of IT projects, especially with respect to Agile projects. The discussions were held around the subjects of agile development, measurement and measurement practices that the respondents saw as meaningful in general for IT projects. The results found initially indicate that value measurement (and also measurement in general) are regarded as essential, but that the application of a standardized framework has not been made due to the lack of expertise on what measures and metrics could be chosen, how would the governance around measurement processes be organized, and how the results could be interpreted to support decision making. A point of improvement that could have been made for the data collection phase is the selection of more respondents, especially from roles with experience in measurement, preferably with the measurement of Agile projects. This was not possible due to the lack of such dedicated roles with the professional network from Shell that we had access to. If such roles were interviewed, we could have obtained more specific data on the appropriateness of the metrics and measures, in addition to information on the measurement processes as a whole. The next chapter of the thesis goes further to analyze these answers and extract key empirical evidence that shapes the agile value measurement framework.

Chapter 5 AGILE PROJECT VALUE MEASUREMENT FRAMEWORK

5.1 Introduction

This chapter develops the value measurement framework prototype, providing the answers to the third research sub-question:

RQ3: How does the agile value software measurement framework look like?

- ***Which existing measures and metrics are suitable, and what new metrics need to be added for measuring scaled agile IT projects?***
- ***How can these indicators be applied in practice?***

The research question is answered by designing and conducting interviews with practitioners from Shell's IT Global Functions unit, investigating the measures and metrics suitable for scaled agile software projects. The answers are matched against existing theoretical assumptions, so that the specifications for the measurement framework can be defined.

This is done in Section 5.2 where data is analyzed through data reduction and comparison with the assumed set of value metrics. Thus, having defined and supported the structural specifications, it follows that section 5.3 illustrates the measurement framework and explains how it should be used in a practical context. The chapter concludes by enumerating the core results of data collection and analysis, emphasizing on the measurement framework. Having described the deliverable, Chapter 6 will follow to evaluate standard scientific quality criteria, as well as reflect on how the framework could be applied within an enterprise-level agile transformation.

5.2 Data analysis and results

The task of data analysis consists of structuring and interpreting the data collected through interviews so that conclusions can be drawn to support or contradict the theoretical assumptions formed in the theory chapter. As exposed earlier, empirical data collected has two core roles. The first is to validate the literature findings and theory based on these findings, i.e. observe practitioner's point of view on the set of metrics and measures selected. The second is to build the measurement framework, by exposing the challenges and potential solutions of agile value measurement in an enterprise context to see what new metrics or practices can add to existing ones. We describe how these results are extracted from the data, and discuss the results as three core specifications for the agile value measurement framework. First, we describe the analysis methods used. Second, we reflect on the data collected with

respect to how it can be split into the three core specifications. Third, each of the three specifications are discussed in detail, to form the measurement framework.

5.2.1 Analysis methods

The unit of analysis is the text that resulted from writing down core points extracted from the audio recording of responses. Because the responses were mostly unstructured, there were no step by step guidelines that could prescribe how data can be analyzed. For analysis, we followed the indications of qualitative data analysis from Research Methods for Business (Sekaran, 2013). Data reduction was used to select, code and categorize the data, afterwards displaying it within a matrix followed by guidelines to use it (which is actually the framework).

First, coding the qualitative data was done through rearranging responses obtained and excluding parts that were off topic. The data was arranged based on the structure of the interview questions, which were used as guidance to form units of code. Therefore, each unit of code (represented by a theme) had to fit the category of a certain question from the interview. For example, one unit of code was related to the theme: professional background of respondents, while another unit of code was represented by the theme: Decision making based on metrics. Coded groups were arranged so that they fit three broader categories that stand at the basis of structural specifications for the measurement framework. We delineated three structural specifications as follows.

First, there is context data, a category which delineates by whom the measurement framework could be used and in what kind of organizational and project development context. This category is built out of the three coding units: professional background and role description, agile/waterfall experience, and project description. These units help to understand who are the usual stakeholders that should be involved in software measurement, what is their experience with both agile and waterfall work methodologies, and what kind of projects they work on. Second, there is the metrics recognition category. This category identifies to what degree are satisfaction, uptake and usage metrics (described in the theoretical section of the thesis) recognized and used by stakeholders. Moreover, novel metrics are proposed in the same category. This helps delineate the contents of the measurement framework deliverable. Finally, there is the metrics application category containing four coding units: (1) “metrics that influence business/financial measurement” for identifying the link between value measurement and the conclusions for business stakeholders, (2) “decision making based on metrics” for understanding what actionable outcomes can result out of monitoring and analysis of metrics, (3) “agile vs. waterfall measurement” for judging the differences and similarities of applying value measurement to the two work methodologies, and lastly, (4) “challenges and solutions for measurement” a unit reflecting on the core challenges of applying a measurement program and potential solutions to these challenges. This final category proposes how measures and metrics can be applied in a practical context.

Thus, the data analysis methods used were aimed at focusing data towards three core areas for agile value measurement: drawing up the context and people involved in measurement, specifying the measures and metrics that are relevant and useful, and considering practical application of measurement along with potential benefits and challenges. The next subsection describes the results of

data analysis, from the perspective of the three categories specified, to validate literature findings and explain new insights into the subject.

5.2.2 Data analysis results – Reflection and Value Measurement Framework

5.2.2.1 Reflection on collected data

The data found is in many aspects supporting the results of literature review and theory building. We will go through the key points that support theoretical assumptions, but also the findings that are complementary or do not match with literature.

First of all, with respect to the application context for utilizing agile development methodologies and measurement practices, empirical data confirms existing literature and adds upon it by highlighting that measurement is difficult because it requires significant planning, implementation and maintenance efforts in order to be of value. Transformation to agile work methodologies seems to be very challenging for enterprise level companies, who (at least in this case) are cautiously adopting agile, and using adapted versions of agile (such as Scaled Agile). This adaptation of agile helps the company maintain its controls and structures in place while arguably benefiting of a more efficient and effective way of delivering IT to internal and external clients. However, as previously indicated in literature by critics of scaled agile methods, we found that there is a lack of enough communication and clear attribution of team roles in the IT project and portfolio environment, with emphasis on lacking business – delivery stakeholder communication. This drawback slows down agile work methods and has strong repercussions on measurement practices that can be applied. Data confirms that measurement programs are rarely implemented due to the fear of increased costs and longer development times, lack of communication between business, IT and clients, and the inertia of traditional development methodologies where measurement practices were not concurrent with development.

With respect to the stakeholders and means of measurement, we found that measurements that are focusing on value and delivery have less to do with technical difficulties but rather with organizational and business issues. This is another point that confirms that implementing a measurement program is difficult form a change management and restructuring point of view. The empirical data made even more evident the parallel that can be drawn between agile transformation and implementing measurement of value. We found support to the assumption that there is an excellent fit between agile methodologies and value measurement practices, because both areas are guided by the similar principles of higher client satisfaction through communication and adaptability. One of the most interesting findings is that value measurement is not necessarily seen as a method to measure agile projects, but rather the other way around, the transformation of agile provides a window of opportunity to adopt value measurement practices as part of all projects in the company, agile or traditional. Respondents reported that tools already exist for collecting most of the measures exemplified in the theory section, so it is mostly a matter of pushing an initiative towards including measurement processes and practices within the software development and application management lifecycle.

Concerning the metrics and measures theorized as useful for value measurement of agile projects, the interviews showed that while most measures were recognized as useful, very few of them were actually monitored, and even less were utilized for decision making purposes. This result supports the assumptions that measurement in general and value measurement in particular is rarely implemented and utilized for project and portfolio level decision making. Also, we have found that the division between business and delivery teams is one of the reasons for metrics not being fully used. While delivery teams do include controls and tools for measurement, they are not responsible (and do not have the resources) to use satisfaction, uptake or usage measurements for drawing conclusions about the application. These tasks remain for business stakeholders. Among the metrics proposed, big data-related usage and satisfaction measures were reported to be extremely valuable if utilized in a relevant way. However, an additional challenge here is making sense of the data, and attributing possible causes to effects. There are no decisional mechanisms embedded in collection tools to reflect what the cause of a certain measurement could be, making this a responsibility of people that need to define causality relationships and decisions that could be taken based on this causality. This means that measures should be made more specific and resources invested in developing causality and decisional chains that are particular to a certain project or situation. Interviewees pointed out that this level of specificity of measurement is not feasible because of too high costs and loss of efficiency in the delivery of software.

Novel metrics and measurement practices have been suggested by only one of the three respondents, in form of AB testing, which is a method where customers offer direct feedback on two or more features, functions or user experiences choices, and then the choices are analyzed to select the appropriate one. This is a proactive method of designing software or features, and can be used for certain types of client-oriented software. However, the other respondents did not provide any novel metrics that they use or have knowledge of, which may signify either that the list we made is quite complete, or that the respondents knowledge of metrics suitable for value measurement of agile is limited. While we do not have empirical proof of completeness of our list, we may induce that the latter option is true, because none of the respondents have been involved in any type of value measurement of agile projects. Interview respondents with this experience would have been preferable, but they were impossible to find in the company context.

In conclusion, data indicates that measurement of satisfaction, uptake and usage with the measures we have collected has a lot of potential for indicating the value of agile projects, but is strongly dependent on the context of application (the project, management styles, the organizational culture and resources of the company, etc.). While this subchapter was analytical and descriptive of the specifications, based upon data analysis and theory, the next subchapter delineates the thesis deliverable, which is a set of prescriptive advice and practices that should be used to implement value measurement of agile projects.

5.2.2.2 Context of value measurement

The first category or specification – context of value measurement – is exhibited in this section. The thesis has set out to understand and give prescriptive advice on how measurement should be done to exhibit the value of agile projects, specifically within an enterprise context. Thus, the thesis results should apply given the following context conditions.

First of all, we argue that the type of company that would seriously benefit from implementing a value measurement program is sized at the enterprise level. This company type will have several IT portfolios to manage (for example, HR, supply chain management, customer management, corporate IT, etc.), each consisting of a multitude of diverse applications. Value measurement implemented across an entire portfolio can help to assess and compare the value of its applications for the business and for target clients. By analyzing measurement results, portfolio-level decision makers can take informed, data-driven decisions on which projects to keep, improve, discontinue, or merge with other similar projects. Another organization-wide context factor that is necessary for value measurement implementation is having enterprise-scale agile methodologies (such as Scaled Agile) already implemented, or in the process of being implemented. As argued previously, plan-based methods do not work well with value measurement. Agile methods are inherently dynamic, and focused on client collaboration, characteristics that are required from value measurement which must be adapted to the concept of value as reported by a client or the business. Apart from this functional reason, there is also a very strong political reason for which value measurement cannot be applied over traditional development. Measurement programs are costly and require significant training, change management, rework of existing processes to work together with measurement processes, and skilled human resources. This is a major change in how a company works at tactical and operational level, requiring executive-level initiative and long-term commitment, and subsequent disposition of project and portfolio management stakeholders to change their work habits and include measurement. Thus, a highly formalized traditional-based way of working is unlikely to accept so much change from a political sense, meaning that agile is not feasible for companies that are too rooted into their current traditional software development methodologies, and formalized organizational structures.

From the point of view of industry or business model, the company adopting value measurement has to have in-house development and operation of IT. Arguably, it makes little difference to what the core business of the company is as long as they control the development of their own applications in-house. Exceptions are companies that work chiefly in services that are delivered through IT in a business to client way, due to the emphasis on value delivered for the end-user or customer. In the case of utilizing mostly outsourced IT, we argue that it would be too complex to implement a measurement program due to control and communication required between the client and the outsourcing vendors. In addition, we have no indication as to what the costs and potential returns of implementing measurement programs on just a few isolated projects (rather than at portfolio level) would be. However, in this case the potential benefit of portfolio level decision making is inexistent because measurement occurs only at project level, with no possibility of comparison. The decisions taken at portfolio level are made based on aggregated data received from project-level stakeholders, responsible for monitoring and interpreting data collected at project-level. Thus, we continue to explain the project level context for value measurement.

At project level, an agile value measurement framework inherently requires that the project is delivered and managed based on agile methodologies. Alternatively, the project should be at least in the process of transitioning from plan-based to agile, or work under an agile-like methodology adapted for project and company specifics. The position of clients of the project deliverable is not necessarily important. The target end-users or customers may be internal to the company, or external. Internal customers should

be considered equivalent with externals because of the novel position of IT delivery within a company, which acts as a service provider. The team or unit responsible of developing and maintaining the software can be considered to act in a competitive environment with potential outsourcers that could make the software. Thus, the business and IT side of a software development project should develop and deliver and measure project value in a similar way for both external and internal clients.

With respect to what roles and responsibilities should be in place when considering value measurement of an agile project, we argue that there are a diversity of roles needed, across several hierarchical levels. C-level executives that formulate strategic directions for the IT function within an enterprise must be involved for defining how measurement should go together with development and what the core objectives of value measurement are. Moreover, support is needed at this stakeholder level for the change needed across IT and their clients, to accommodate a new measurement program in combination with agile development methodologies. Moving one step down the hierarchical level, we argue that IT portfolio managers and business owners of the functions supported by IT are responsible for establishing how the measurement goals set by C-level executives should be applied in their particular cases. For example, on the one hand, an HR director requiring software to be delivered by IT should also establish the value targets (KPIs and metrics) and the measures that should be followed to assess them. Portfolio and project managers in the area can then decide if these measures can reflect the KPIs imposed based in their specific project contexts, and aggregate data and conclusions upwards. On the other hand, there should be counterparts on the IT delivery side that can assess if the KPIs and subsequent measures can be collected, what costs and changes they would incur for the project timelines, and proceed to implement the controls and tools needed for measurement, as required by business. We refrain from detailing a more explicit map of stakeholders and their relationships, because this is highly dependent on company organizational structure, and furthermore on the team structures of IT delivery and business units. We conclude by emphasizing that the roles for guiding measurement do not need to be highly specialized in measurement practices, but rather understand business applications of software very well, and be able to delineate customer and business value objectives and what indicators could measure them. A decisive factor for this to work is effective communication and collaboration between the different stakeholders involved in development and measurement – IT delivery, business demand, and potential end-users of deliverable.

Having defined the context of value measurement, we continue to suggest the contents – the metrics and measures that can be tracked in order to assess an agile project.

5.2.2.3 Contents of value measurement – metrics and measures

This subsection illustrates the set of measures and metrics, categorized by what they can potentially measure – satisfaction, uptake and usage. The metrics and measures from the theoretical chapter remain in essence very little changed, as the empirical data analyzed confirmed the usefulness and relevance of the metrics, and indicated that the greatest challenge is not finding new metrics but deciding on a set of relevant metrics contingent on context, and applying them meaningfully so that conclusions could be drawn on the value of the software. Therefore, the major contribution of this research is in proposing how the measurement framework could be used, by what stakeholders and in what context, and what are the greatest challenges to block measurement programs.

Measurement Type	Metric / KPI	Measures	Description
Satisfaction	Feature appropriateness	AB testing	Choice between two or more mock-up or proof of concept features directly by potential users.
Satisfaction	Fulfilment of client needs	Number of change requests per requirement	Fewer change requests and less discarded requirements indicate that client needs are fulfilled, therefore indicating satisfaction with delivered product as it is.
Satisfaction	Feature Quality	Number of issue reports submitted by users	The metric exhibits the quality of delivered features by comparing the number of user issue reports across similar projects, over specific time intervals within project development.
Satisfaction	Customer satisfaction with product	Quality of product, price adequacy, reliability in terms of time and costs, completeness of product delivered at the end of each sprint or release, flexible handling of changes in requirements, good collaboration with the development team	This collection of measures targets business and user satisfaction with a developed product. Measurements are made through surveys conducted at the end of each sprint/release of a feature, or entire software.
Satisfaction	Customer needs and hidden requirements	Big data: Helpdesk tickets, feedback from support interventions, and topics discussed on user forums	These measurements are classified as VoC (voice of consumer) methods because they measure hidden requirements and customer needs by assessing their helpdesk requests, communication with support, topics of interests on user forums, etc. The big-data associated measures can help detect spikes and problems that are not related to technical faults but with the learning curve of users.
Usage	Feature usage	Relative actual usage of features by end-users, intended usage by developers, actual usage threshold	These measures assess if a feature is underused or misused by end-users by comparing them to usage thresholds and intended developer usage. Usage below a certain threshold indicates if candidate or existing features should be removed or improved.
Satisfaction	Content & Usage Analysis	Adjusted Ideas (number of ideas about features, functionality and software)	A high number of (user) ideas implemented reflects current, and predicts future satisfaction of users.
Uptake		Intensity of collaboration (number of users interactions with each other based on software content or functionality)	Reflects value derived from collaboration and communication of various users by making use of the software.
Uptake		Degree of cross-linking (range of users linked together)	

Measurement Type	Metric / KPI	Measures	Description
Usage	Business Value indicated by usage	Log-in number and frequency, sequence of user steps	Quantitative measurement of database and logs on how much and how the software is used over time. Intensive and growing usage of software should indicate that it is valuable for both customers and business
Usage		Feature number of times used, frequency, duration	
Uptake		New users gain rate or unique visitors	
Usage		Hits per time period	
Usage		Number of sessions, duration, average time per visit, frequency	
Satisfaction	Usage Value: User interviews and surveys	User requirements	Qualitative assessment of how users perceive that their needs are met, and that the software is easy to use in given use cases and following the most common usage behaviors. Data is usually collected by business stakeholders through interviews and surveys with the end-users
Usage		Usage behavior (sequence of steps, inputs, etc.)	
Usage		Use case validation	
Satisfaction		User satisfaction with tools and processes	
Usage		Applicability of tools / knowledge of users about their applicability	
Satisfaction	Business Value through satisfaction and efficiency	Effort of utilizing the software/tools (through user Interviews & Surveys)	Assessing the effort for utilizing software and associated tools by end-users indicates user satisfaction and acceptance. Low effort means high satisfaction and efficiency of users, which creates business value.
Uptake	Reusability	Was the application developed to meet one or many user's needs?	These metrics are part of the plan-driven measurement practices, and are useful for assessing if software can be extended to multiple from functional and compliance points of view.

TABLE 5.1 UPDATED VALUE MEASUREMENTS BASED ON INTERVIEW RESULTS

Table 5.1 contains only one additional measure and metric (AB testing) added based on empirical results. Also, there are minor modifications to the other metrics that we have found from literature. We argue that the reason for this is that the gap is not in the absence of measures and metrics, but the lack of practical application and empirical evidence on the effectiveness and efficiency of their application. The collection of measures and metrics found can be used as a basis for establishing a measurement program, through discussions on which of them should be used, how, and for what purpose. This subject, applicability and usage of measurement is treated in the following section.

5.2.2.4 Applicability of measurement

As we have seen, the core challenges of agile project value measurement have little to do with technical issues, but a lot to do with inclusion into the software development lifecycle and application management lifecycle. As found in theory and from the data collected, the measures are mostly known and desirable and the goals of satisfaction, uptake and usage can be called standard for any

measurement program. However, their application (metrics chosen and methodology to apply them) differs based on the context of the software project and its stakeholder's knowledge, resources, etc. In this section, we propose how agile project value measurement should be planned for, who should be involved in a measurement program and how it should be intertwined within planning, development and operation stages of an application's lifecycle.

The first step should occur during the planning stage of a new (or to be modified) application. Value measurement as a process that works in parallel with development and operation should be brought into discussion. The goals for measurement should be set around evaluating the usage, uptake and satisfaction of a software and its features, so that decisions can be made at both project level (the features that should be enhanced or dropped, trainings required for user learning, etc.) and at portfolio level (which projects should be merged, discontinued, improved, etc.). Afterwards, roles and responsibilities should be distributed amongst both technical and business stakeholders to track the progress towards the goals set, and the relevance of goals over time. Ideal stakeholders for these roles do not have to be highly technical-oriented, but rather have good knowledge of user requirements and application lifecycle management, but also some knowledge of the software development lifecycle (for example: knowledge of agile processes, resources needed for feature changes, etc.). Appropriate positions must be from both delivery and business sides: business analyst, product owner, project & portfolio manager, software architect, and the business-side stakeholders of a product. The most essential part for establishing a plan for agile project value measurement is the collaboration between these stakeholders, especially between business and delivery teams. This collaboration consists on the one hand, on business stakeholders being constantly updated and informed about the resources needed for measurement and the technical possibilities of implementing data collection mechanisms, and on the other hand, on delivery team stakeholders being informed about the what the project value targets are and what their progress is (based on the measurements interpreted by business).

Upon deciding the measurement program goals and responsible roles for achieving them, metrics should be chosen to help fulfill these goals. The choosing of metrics can be done by a joint effort of delivery and business teams, based on the list compiled in this thesis. Apart from the project stakeholders, if the measurement program is intended to be used at portfolio level, there should be additional effort in choosing metrics that can be valid throughout the entire portfolio. This increases the complexity of measurement, as there is need for different project owners to collaborate and align their measurement efforts. However, without a standardized measurement program, portfolio level decision makers cannot take informed decisions on projects because measurement data meaning cannot be extrapolated above the project level. While metrics should be decided on both portfolio and project level, their composing measures have to be specific to the project level. This means that each project team should choose their own way to measure things as long as they follow the standard measurement goals (the metrics). Just as with metrics, measures should be chosen by a collaborative effort of software delivery and business stakeholders. Business stakeholders can validate if metrics are actually useful and meaningful for estimating a metric, while delivery stakeholders can evaluate the resource cost and technical feasibility of applying certain metrics. There are a few exceptions in the cases where there is no need of the software delivery teams for implementing certain measures, for example, in the case of interviews and surveys for evaluating certain functional characteristics.

After deciding which measures to follow, how it should be done, and by whom, the initial planning part for measurement is completed. However, the major difference between agile value measurement and traditional measurement practices resides in the actions done after planning, and during development and operation. Just as with agile development, measurement in agile should be continuously improved to better fit changing client perceptions of value. This means that the measures and metrics established in the planning phase are not definitive, and there will be no "final" version of the measurement program and its characteristics. Changes in the measurement program should be aligned at project level with changes and decisions taken after agile sprints. More specifically, stakeholders should be critical on how easily measurements were made (in terms of effort and time), how well could data be communicated and interpreted by business decision makers, and how well can data support project and portfolio level decision making. Thus, by aligning the management of agile development processes with measurement, they can be controlled simultaneously. A more specific exemplification would be how a requirement (or user story in the case of agile) is managed during development. Apart from the regular tasks associated to requirement management such as development, code review, test creation, test execution, UX design, etc., one or more measurement tasks could be included.

Furthermore, we argued that measurement is required all throughout development and operation. This means that after release, measurement continues to collect information about how end-users utilize the software. These tasks could be included in the regular maintenance and support processes of the software. In this stage of the application lifecycle, the measurement responsibilities previously assigned to the delivery team pass on to operations. However, the delivery team will still have the role of acting upon the interpreted data from measurement, with actions such as modifying features of the software, working together with operations teams to improve application performance, or releasing updates and fixes.

In conclusion, we argue that the measurement of agile project value should be intertwined with processes that normally take place during the stages of project development and operation. The selection of metrics and measures can be done based on the collection of satisfaction, usage, and uptake metrics and measures depicted in Table 8. The specific metrics and measures that should be used are dependent on the type of project, the structure of its team and relevant business, delivery and operations stakeholders, and on the guidelines decided at portfolio level, for implementing a standardized measurement program across multiple projects. The choice of metrics and measures should be adapted over time as the agile project evolves and is released, meaning that the collaboration between the various stakeholders should continue during the entire application lifecycle. Generally, we could not distinguish which measures could be attributed to which specific lifecycle phase, but for pre-release phases, feature appropriateness metrics such as AB testing could be used for better matching software functions and design with user requirements, whereas for post-release, or during beta-testing of the software, metrics on feature usage, customer satisfaction or business value indicated by usage can be useful for adjusting the software on long term.

5.3 Chapter conclusion

This chapter has presented the main findings of the thesis by analyzing empirical data which supports the theoretical assumptions made, resulting in the agile project value measurement framework. This final subchapter reiterates the key points of data collection, analysis and results.

Data collection was done by means of three semi-structured interviews. The aim of the interviews was to inquire on how measurement is and should be done to assess the value of software projects for their clients (business) and end-users. The context in which the software projects are developed and measured is in-house, within an enterprise setting, and usually within a broader portfolio of projects that serve the same function. Interview respondents were selected so that they have experience with both the demand (business) and the supply (delivery), specifically focusing on projects developed with agile methodologies. Discussions revolved around a set of metrics and measures (established in the theoretical chapter) for assessing the value of software projects, by means of measuring satisfaction, usage and uptake of a software. Apart from discussing the measures and metrics themselves, a prominent discussion point that was uncovered through interviews was about the politics and organizational context that is required for successful value measurement.

By coding the responses based on core themes followed in the interviews, we structured the information obtained. We found that while most of the metrics and measures were not new to respondents, and could be technically collected, they were mostly unused and difficult to interpret. We found that the most challenging side of implementing a measurement program (value or otherwise) is not necessarily in the choice of the metrics and measures, but in setting up and maintaining the right context throughout development and operation of software. Based on these findings, we argued that a value measurement program should be operated simultaneously with agile development and operation stages, and that stakeholders should have clear measurement responsibilities linked to their application management responsibilities. We presented a few high level steps of a value measurement program should be linked with development, and described how delivery, operations and business stakeholders should collaborate in the planning, implementation and continuous adaptation of value measurement, to fit with project needs and characteristics, as it progresses in its application lifecycle. The next chapter presents an evaluation of scientific methods and practical relevance of results.

Chapter 6 EVALUATION/ VALIDATION

6.1 Introduction

At this point in the thesis, the deliverable has been presented and the research methods to attain it explained. However, the measurement framework has an inherent practical orientation, determining the need for reflection on how well it can be applied in practice. Moreover, because it is out of the research scope and possibilities to observe how the framework would be applied in an actual use case, we decided to evaluate the process of designing the deliverable. In this way, we can at least assess how the results of this thesis were reached, and how they would potentially be applied in practice, in the absence of a more applied opportunity for evaluation.

Therefore, this chapter aims to answer the final research sub-question:

> ***RQ4: How does the measurement framework fare against practical and academic evaluation?***

To answer this question, we first, assess the scientific methods and quality of the research process used to develop the thesis deliverable, to support that the results are significant and reliable, and thus valuable for both scientific and practical communities. Second, assessing the practical relevance and potential value through reflection, supported by evidence drawn from interviews and knowledge gathered from literature review.

6.2 Scientific evaluation

6.2.1 Objectives

The objectives of the scientific evaluation relates to assessing standard research quality criteria. Also, it is important to distinguish what kind of research quality criteria apply, contingent to the subject under evaluation. For each of the stages of research – goal definition, research question formulation, literature review, theory building, and data collection and analysis – we follow the quality criteria and methods that are relevant. These distinctions are made based on Sekaran, 2011, and Verschuren and Hartog, 2005, where evaluation methods are discussed in general and in particular for practical-oriented research. Thus, the objectives of these subchapter are to identify the relevant evaluation methods and see how they can be applied to assess quality criteria.

6.2.2 Scientific quality evaluation methods

The first step for scientific evaluation is to define the object of evaluation and its particularities. Because we are discussing about a practical deliverable, it is mandatory to assess its feasibility for practical use from the beginning of the design process and all along until the end of delivery, and not only after the "product" has been delivered. Verschuren and Hartog (2005) support both ex-post and ex-ante evaluation types for the case of practical deliverables in research. This is because in the practical world,

requirements change often, demanding that any design-oriented research is flexible to adapt. Much like with the agile software development and testing process that we touched in this thesis, evaluating a practical deliverable requires stages similar to sprints, for evaluating the gap between present and to-be state of the deliverable. Moreover, again similar to agile practices, evaluating a deliverable before its fruition determines lower effort and costs for changing it after it is done.

This focus on ex-ante evaluation can be reflected throughout the thesis in how the research stages progressed. Many times throughout the research design process, we have turned back towards previous stages, such as goal formulation, and research scope definition, based on new insights received from industry. Because research was done within an internship, the impact of new insights was strongly felt, sometimes even frustratingly so. However, despite the blurring of the different stages of research and overlap, we still have to focus the evaluation of each of the stages. To structure this, we choose to split the design phases into plan, process and product evaluation, following the design cycle of Verschuren and Hartog (2005). The process and product evaluation are out of scope, since they involve following a case study approach and actual modifications of the deliverable based upon observing and measuring application. Therefore, we characterize our research by the plan stage, on which we apply plan evaluation. Plan evaluation involves verifying if the design planning is relevant and carried out correctly. Since our deliverable has not reached a live environment, we assume that it can only be evaluated by plan evaluation methods. Therefore, we apply a logical and empirical verification on the stages sub stages of our research.

6.2.2.1 Research design stages evaluation

The first stage under evaluation is the goal definition stage, where we verify how the research objectives have the necessary quality. We look for the purposiveness of the objectives to reflect objective questions that are meaningful for both scientific and practical stakeholders. We reflect that the purposiveness of the objective is strong because of two reasons. Firstly, the scientific community has not yet sufficiently addressed the problem of quality measurement in agile practices. This is a serious gap because agile methodologies have become the dominant way of developing software, and measurement has been recognized as an essential part of ensuring that the products of agile development are valuable for clients and produce the necessary returns for investors. Secondly, the practical side that consists of software development companies or corporations that develop software as a non-core function prove to be significantly challenged by measuring the value of software, and assessing how agile practices benefit to this value. Thus, we judge that the objective of exploring and delineating agile software measurement practices and metrics is extremely purposive for research and practice.

The second stage under evaluation is the theory building section, which analyses literature findings to create a foundation for answering the research question. Here, we are interested in objectivity and rigorousness in considering a broad scope of literature and research. For selecting the articles for review, we have gone through several areas related to software engineering, methodologies and software measurement. The sources selected were not only based on the relevance of the subject, but by the perceived quality of the paper, judging by number of citations, and the notoriety of the journal or book publisher. Moreover, we followed to avoid searching for articles that only praise agile

methodologies and value measurement without considering that these concepts cannot work without the classical methods to complement them. In this way, we argue that we achieved minimal bias towards finding the sources that favor our predicted research directions.

The third stage under evaluation is the data collection. Here, we auto-evaluate factors such as the appropriateness of the data collection methods and the methodological rigor of interview design and carry-out. We argue that the semi-structured interviews structures were appropriate because of the level of maturity of this subject. Software measurement in agile is a novel subject, but it is not too innovative or focused on niche areas of knowledge. Thus, an unstructured interview approached would arguably have been too explorative and broad, because there already is some information and guidelines on how value measurement should be done based on classical measurement practices, and based on non-scientific experience of practitioners. On the other side of the coin, a structured interview would have been too restrictive for letting the subjects express their opinions and perceptions of value measurement. This would have narrowed the scope of results too much, possibly blocking new insights. With respect to methodological rigor of carrying out the interviews, we assess that this characteristic may have been improved. At times, the questions and protocol were not respected and deviated too much from the subject due to lack of experience of respondents. This denotes also that the interview questions could have been better tailored to the background and experiences of respondents.

Fourth and last, we evaluate the analysis of data and design of the deliverable based on it. This stage of evaluation follows that information is coded into the right categories and that the inference of conclusions based on these categories is characterized by internal validity. We judge that the categorization and coding was done correctly, because it was strongly following the guidelines defined in the theoretical chapter. Mainly, we formed the interview protocol based on literature findings, thus predicting already what categories will emerge. Furthermore, the categories had the purpose to support the theoretical assumptions about the value of agile measurement and value measurements and metrics applied in enterprises. Thus, we can say that this relationship between the theoretical assumptions, interview definition, and data analysis supporting assumptions brings positive evidence for the internal validity of results.

6.2.2.2 Deliverable evaluation

Except for the evaluation of the stages by a few standard research quality criteria, we also look into the broader scope of the entire research results, mainly, into the generalizability and reproducibility of studies. We argue that the generalizability of the results is uncertain, despite the fact that the results seem easily applicable for a broad number of cases. The measurement framework offers a set of value measurement metrics and practices that must be applied to certain situations, and for each one differently. We did not research how these metrics could be tailored to the software project particularities but we expect this need for customization to be significant. This is because in practice there are constraints that affect how organizations apply such measurement frameworks such as budget, knowledge and human capital, change management barriers and so on. We argue that it is difficult to obtain high generalizability for a measurement framework because there is a gap between theory (i.e. the metrics contained in the framework) and practice (their application in a live environment, where there are many variables and requirements for customization unaccounted for in

this study). We argue that future research should dive deeper into value measurement of agile methodologies and distinguish between how sets of metrics can be appropriated to various cases.

We judge the reproducibility of carrying out this study to be high, because there are a prescribed and objective set of steps that can be followed. First of all, the metrics and measurement practices and how they were decided upon has been documented and referenced, making it easy to trace back to the sources. Also, the metrics and measures used are specified, and an interview protocol for a different company could easily be made based on these metrics and the target goals that want to be achieved with the respective data collection methods. A side which may challenge reproducibility is finding roles that are similar to the ones interviewed in data collection. Because there are very few companies that have roles dedicated to software measurement, and because there is no standard way to measure agile value, it would be difficult to find the same answers in other contexts if other researchers would apply similar interviews. Another quality factor related that is negatively affected by this issue is the objectivity of the interview respondents, and subsequently of the results achieved. Despite the fact that the measures and metrics were established in an objective manner, we argue that there is considerable bias and subjectivity due to this restricted range of respondents. However, specifically due to the fact that we recognize this weakness, it is recommended that further research should drill into how the measures and metrics found in literature actually fare in practice, and prioritize, reduce or enhance the list and practices presented in this thesis.

6.3 Practical evaluation

Even if an evaluation of the deliverable in a practical case study is out of scope for this thesis, it is important to at least reflect on the potential impact, benefits and challenges of applying the software measurement framework. This reflection is done based on what inferences can be made based on data collected from interviews and based on the general experience of the author as an intern at Shell Netherlands.

Based on the general reception of the topic with the people discussed formally (interview respondents) and informally (other Scaled Agile project and program stakeholders), the following points can be made. First of all, if not all the metrics and measures contained in the framework, most of them were considered useful for better evaluating business value of software projects, for reporting and decision making at IT project and portfolio level. On the downside, there was little to no feedback on the implementation of these measures, which reflects that further steps need to be taken in research and practice to close the gap between the definition and implementation of value measurement practices. Also, from the difficulty to find respondents that could have given more precise answers to the interview questions, we judge that there would be a serious lack of resources to assign to measurement, data collection, analysis and reporting tasks.

Second of all, based on the motivations that we exposed in the Problem statement section, we can assert that stakeholders in project and program IT development are interested in judging whether the Scaled Agile methodology works better for the company when compared to previously used methods, and where it would work best based on project topic. It is possible that through measurement of project value, inferences can also be made on the work methodologies. Thus, although the measurement of

methodology effectiveness is specifically out of scope for this thesis, it can be considered as a byproduct of project measurement, because of the arguable link between the value of a methodology and the value of the product it creates. Again, the downside of applying the measurement framework here is that this framework focuses on agile development, and was not applied previously on other projects, and therefore has no term of comparison. For this type of measurement of methodology value, a similar set of metrics and measurement practices should be applied on long term, for similar projects, that are executed with different methodologies. These conditions are extremely improbable to occur, especially in a company that does not have IT development as its core focus. Therefore, we can at best say that the measurement framework exposed here can provide some guidelines and foundations for assessing agile vs. waterfall project methodologies, but is far from a definitive practice that can be applied to compare the two.

Third and final, despite its obvious implementation gaps to fill, and resource barriers to surpass for being applied in practice, the agile project measurement framework would be valuable in practice because it has a lot of potential to cover areas that are very meagerly treated. In the business environment of managing the IT function at program and project level, we have found that at least in the case of this study, decision makers often lack hard evidence and must rely on historical data, subjective perceptions, or recommendation from external parties to decide which projects are valuable and which should be terminated, improved, or merged together. This situation backs the supposition that such a measurement framework would be beneficial for decision making because it would offer some (even if not exhaustive or fully correct) data. This data could be aggregated from project to program level, and then further to C-level executives that can reconsider the entire strategic direction of the company IT from something else than just the classical financial metrics, and market, technology or competitor analyses. However, we stress the fact that there are many challenges to applying such a measurement practice and that the data it produces should not be taken directly as solid and flawless evidence for decision making.

6.4 Chapter conclusions

In this Chapter, we have approached the evaluation topic from a scientific and practical perspective, analyzing several standard research quality criteria for both the process and the product of this research. Furthermore, we briefly reflected how the value measurement framework would be used in the practical environment of Shell's Scaled agile program. We began by specifying that the evaluation refers only to the plan stage, following Verschuren and Hartog, (2005), and that it is done ex-ante, because we do not yet have a finished product that is applied in practice. The research design stages were evaluated against relevant criteria for each, finalizing with an evaluation of the deliverable.

Chapter 7 CONCLUSIONS

7.1 Introduction

This final chapter recounts the core achievements of this thesis, and describes the steps made to realize them. This thesis has sought to explore how IT project and program level decision makers can better evaluate software development projects through value measurement. More specifically, how agile projects can be measured so that their benefits, costs and risks become more transparent. Furthermore, this objective was sought in the context of large enterprises that move from traditional to agile development methodologies. Due to this practical environment, the research was aimed at not only exploring but distilling a deliverable which may help business and IT stakeholders from companies determine what measurement practices, metrics and measures they can use for agile project measurement. Thus, the research objective was to:

> ***RO: To deliver a set of metrics, measures and recommendations, targeted at IT project stakeholders to assess the value of agile software development projects in an enterprise context.***

The objective was fulfilled by answering a set of research questions and sub questions, which have been answered concomitantly with the research flow structured into chapters. The following sections of this chapter describe this flow and underline the main achievements for each of its components.

7.2 Research flow and core results

The thesis began by outlining the research context and problem, which is rooted in the shift of large enterprises towards more agility and control through insourcing. Through scaled agile, large companies can potentially benefit from the arguably superior agile development methodology that emerged at the beginning of the new millennium. However, agile development brought with it challenges, such as the blurred borders between business and IT, which require higher integration and understanding between the two groups. An additional challenge that is meagerly treated is measurement, to understand, track and estimate the value of agile projects and their deliverables. The issue that we came up against is that there are no measurement methods to evaluate value of large scale agile projects and portfolios, because measurement in the agile software development area has only scarcely been formalized in research, being mostly represented until now in practitioner whitepapers or case studies on specific contexts. Thus, the study relevance is high for both scientific and nonscientific communities. For researchers, this study proposes a step further in measurement of software and complementing existing practices with value-oriented metrics and measures suitable for agile. Societal relevance lies in IT project and portfolio level decision making based on data other than purely financial or technical, but focused on value delivered to the client and to the business.

Upon establishing the grounds for the problem, the main research question was sketched to depict how a value measurement framework looks like and what are the implications for stakeholders of agile IT projects. The question was answered by splitting it into sub questions and answering them throughout chapters in the study by following the design science method presented by Hevner (2007) - the three-cycle view. We explained how the relevance cycle was followed to identify the domain of application and practical context of the problem, the rigor cycle to draw upon the knowledge identified and build a foundation for generating a novel deliverable, and finally the prototype and evaluation, where through several iterations of reshaping the framework, we reached a prototype state which could be evaluated from a theoretical and practical perspective. Each of the cycles answered a sub-question, structured in the thesis through its chapters.

7.2.1 Domain description and theoretical foundation

The domain description chapter answers the first research sub question on the implications of adopting agile practices on IT project development and measurement, in the context of large companies. Literature review was conducted to answer the bulk of this research sub question. We began by explaining how Agile developed and reached a mature stage as the dominating software development methodologies, overtaking Waterfall. Agile software development process models better fit the shifting market requirements and volatile technological changes of today's business environments, due to their lightness in defining requirements, focus on stakeholder and client communication, and periodic readjustments in between the testing and development "sprints". We also identified and discussed the challenges that come with this new way of working, especially for large enterprises which are rooted in the heavily controlled and procedural waterfall methodologies.

We identified that one of the challenges in this shift is to evaluate the benefits of Agile projects, and of agile over waterfall in general, especially when there is more than one project that makes this shift. among which software measurement is underlined. To appropriately support decision making on project and portfolio level, data is needed from measurements of more than the project efficiency and performance, but on its value. However, we saw that software measurement is often difficult to implement, because although it has potential high rewards for decision makers, it inflicts a high cost of resources and effort that has to be invested and maintained on long term for results to be visible. Nevertheless, measurement is necessary if decision makers want to have reliable sources of data to base their IT decisions on, and value measurement fits well with agile methodologies.

After explaining the implications of agile methodologies and their measurement, we go into building the theoretical perspective, which answers the second research sub question. Here, the core software measurement literature was reviewed and conclusions were aggregated to underline how software measurement was done and what are its future directions to align with agile development practices. We distinguished between the main goals of software measurement as being efficiency, effectiveness and value, and argued how the former two are traditionally dominating goals in waterfall project measurement, while the latter, value, grows to complement the others for assessing agile projects. To measure value, we divided it into three core components: satisfaction of end-users with the software solution in terms of ease of use and fit with their requirements, uptake of the software in terms of breadth of users and growth of adoption, and usage of the software in terms of who accesses it and in

what ways (user behavior, features used, etc.). The following step we took for understanding how to evaluate value was to match its three components with potential metrics / indicators, and their corresponding measures. These metrics and measures were selected based on a further round of literature review digging deeper into the scientific articles that study value measurement-related topics. Thus, we were able to distill a set of metrics and measures which we assumed as potential candidates for reflecting agile project value. This is the theoretical foundation of the thesis, and was followed by an empirical assessment, to understand how our assumptions fared against application in practice.

7.2.2 Core specifications and the agile value measurement framework

It followed that the next research question could be answered, showing how the agile value measurement framework would look like. This chapter was characterized by data collection and analysis. We collected data from IT project and portfolio level stakeholders with Shell's IT organization, which underwent a shift towards Scaled Agile methodologies. We used semi-structured interviews to address a set of questions on the measures and metrics established in the theoretical foundation. The objective was to find if and how the metrics are used, and what potential they have for measuring agile project value. The three respondents selected were able to give high level answers, but lacked in preciseness, indicating the scarcity of knowledge and focus on agile software measurement. However, this lack was strongly complemented by a positive and enthusiastic attitude towards implementing value measurement in the future, ascertaining that this type of measurement would be a decisive factor to improve decision making, and better understand which projects were important for the company and clients and which are not. Following a structured analysis of the data obtained, we managed to code the qualitative data by removing off-topic areas and categorizing it into three structural specification for our framework.

First, we delineated the context of usage for the measurement framework, specifying by whom it could be used, in what kind of organizational and project context. We found that transformations to agile work methodologies is specifically challenging for enterprise level companies who are cautious to adopt agile and use adapted versions, to fit their controls and structures. We also found that software measurement is scarcely done in a standard way at enterprise level due to lack of resources, communication and interaction between the business and the IT delivery stakeholders. Despite all groups recognizing their worth, software measurement is blocked by the costs and risks perceived, and the lack of understanding of how to design and implement a measurement program

Second, we distilled the specification regarding the means of measurement, or its implementation. We found that implementing measurement focusing on value is strongly related to agile transformation, due to the fit between the two in their guiding principles - targeting high client satisfaction, supporting adaptability, and requiring cross functional and cross-hierarchical communication. Moreover, we have found that the challenges with implementation are often unrelated to technical aspects of measurement (i.e. data collection through automation and programmable tools) but driven by change management and restructuring challenges. Despite the existence of tools for measurement, a measurement programme must be pushed forward, championed and included in the software development and application management lifecycle by managers and non-technical people who have the political strength to do so.

The third and final specification refers to the metrics and measures theorized as useful for value measurement. Data shows that while most measurements were recognized as useful, very few of them were actually monitored or utilized for decision making purposes. Again, the tools and controls to collect them existed, but there was no drive to start and coordinate a measurement program. The most popular metrics and measures were identified as related to big data analysis, and client-satisfaction, said to be extremely powerful in evaluating not only the present solution but predicting what directions should be pursued. However, it was recognized that the analysis and interpretation of such data is extremely difficult. There are no decisional mechanisms to define the causality between collected data and recommended decisions, and this is seen as a further point that blocks implementation of a measurement program.

Based on these specifications we distilled and presented the agile value measurement framework through a table that categorizes metrics and measures, and explains the context in which they are useful. This table was followed by a section detailing applicability advice for measurement. A core recommendation for practitioners is that value measurement should be taken up as an inextricable part of the software development process, all throughout its lifecycle and stages. Moreover, it is essential that responsibilities for executing, coordinating, analyzing and reporting measurement activities and results is done by both business and technical (IT delivery) stakeholders, and shared between different functions. This involvement of various stakeholders is especially important at the design phase of the measurement programme, for selecting the appropriate metrics, measures and practices, and understanding what conclusions could be drawn based on the analysis of these metrics. Last but not least, agile measurement should be adapted as the project progresses to reflect the changes in the project itself. This means that measures, metrics and practices should be changed to better reflect the clients' value expectations from the software. It is also important to mention that such a measurement program needs to be upholded on long-term, so that historical data is created and set as benchmarks for future measurements, and for various projects in the portfolio.

7.2.3 Evaluating the research process and deliverable

Following the definition and recommendations for the agile value measurement framework, we went into evaluating the research process and its result. The evaluation is considered to be made on the plan stage based on Verschuren and Hartog (2005) because the study has not reached the point where it can be observed in a practical context or case study. The evaluation was done through reflection against research quality criteria assessed based on the stage the research went through.

For the goal definition stage, we concluded that there is strong purposiveness in the research objective because of both the scarcity of research on this subject and the practical need to address the challenges of agile software measurement, so that businesses can be more effective with the software they develop for internal or commercial purposes. With respect to the evaluating how theory was built, we reflected on the objectivity and rigorousness factors, and judged them to be well established. We argue this through the several iterative steps taken for reviewing literature to include sources that are not biased towards just traditional or just agile measurement, but combine both. Regarding the data collection, appropriateness of methods and methodological rigor of interview design was pursued. We reasoned that while the semi-structured interview methodology fitted the research objectives and

context, there could have been more rigor in conducting more precise interviews and selecting a high number and more specialized range of candidates. Considering the data analysis and design of the deliverable, we argued that categorization and coding was done meaningfully thanks to the strong guidelines obtained in the theoretical foundation, where it was outlined what value measurement could mean and what factors can help determine it. Finally, we evaluated the results themselves, as being difficult to generalize because of a significant difference between theory (the measurement framework) and practice (how it would be applied). Whereas the metrics and measurements' benefits are visible in theory, there are several constraints that create uncertainty of these benefits when measurement is applied in practice (budget, specific knowledge and skill for applying measurement).

7.3 Limitations and future research

As exposed in the evaluation sections and all throughout the thesis, we recognize the limitations imposed on this research, due to multiple factors, such as the limited applicability and verifiability of the deliverable in a longitudinal case study within an enterprise context, the difficulty to go into more detail with metrics and measures without a context to determine causality and usefulness of these metrics and measures, and the thinness of a dedicated area in business and IT for measurement. This section also servers for a reflection on the entire process of writing the thesis, and on future directions that we see fit.

The first limitation we recognize, and arguably the strongest to deter progress in the area of value measurement of enterprise level agile projects is the difficulty to apply research and theory into practice. Because usually measurement was found to be secondary and as such, mostly excluded from many of the steps of an application/software project lifecycle, there is little drive to implement theoretical frameworks within running projects. This is a major problem because we observed that high integration is needed between development, measurement and business, so that the measurement results can be considered reliable and useful for decision makers. And thus, based on the data we collected and on our perception of the problem, we see it more as a political issue where measurement is too weak in the political decision making arena, despite its recognized potential benefits. However, we reflect that this problem could be minimized is measurement experiments are made within pilot projects where risks are lower, and where there might be more acceptance for trying novel and innovative ways of working, on both development and measurement sides. Also, we judged that a limitation specific to the context of this research is that within Shell, there is already a high workload in practice for agile transformation, and there is little space and resources for adding another dimension of change in terms of measurement. Thus, apart from applying the framework in a pilot project, the overall company should also be more experienced in agile work methodologies so that they do not fight on too many fronts at the same time.

The second limitation we bring up is the difficulty to go into more depth with operationalizing metrics and measures. This limitation also exists in agile development, and is arguably a strong reason for why agile methodologies do not have detailed steps for implementation, but rather focus on principles, best practices and recommendations. We have found that existing research is cautious to recommend one metric over another, or a preferred way of applying measurement over another. In one case, we have

even received a refusal from the author of one of the articles read, who denied our request to further detail the metrics and measures he exposed in his paper, due to the danger of taking these directly into application instead of tailoring measurement. Thus, we follow the same direction and support the fact that these metrics, measurement practices and measures exposed here (and elsewhere) should be taken only as guidance to develop a customized measurement framework depending on the project that is to be measured.

Last but not least, empirical research in the area of value measurement of agile projects is limited by the lack of dedicated roles, or experts in any of the measurement related activities: measurement planning, implementation, interpretation, and extracting conclusions. As we have found out, measurement responsibilities are passed on depending on availability and not depending on specialized skill or experience with this area. This means that there are many understandings of what measurement is and how it should be done, and that there is no single source of truth to reflect this area in a project / portfolio context. We support this limitation based on the empirical findings from interviews, but also from the lack of literature evidence towards any kind of formalization attempt of measurement-related roles and responsibilities.

Finally, we consider that continuing to research this subject has many opportunities in both depth and breadth. As mentioned before, it is essential that research is done to observe if the results found in this thesis apply in other company contexts and what complementing metrics and measures might appear. Also, longitudinal studies should follow to understand how value measurement practices studied actually fare alongside development, in a practical case study or experiment. Thus, we consider two core future research questions.

(1) How can an agile value measurement framework be applied as an integral part of an agile development project in order to predict project value during development and continuously monitor value upon release into production?

This research question can be answered through a practical case study where the researcher could work in parallel with key agile project stakeholders in order to implement the measurement framework proposed in this thesis. This case requires a very high involvement of the researcher possibly based on a research partnership with a company that would want to try out a pilot for agile software development and measurement. The potential results of such a research would be of great value to get insights into how the metrics and measures proposed apply, i.e. how would data be collected, by whom, how would it be interpreted and by whom, and how would the final results of measurement analysis be used to support decisions on agile project direction, continuity, or interruption. These results can be used to validate and refine our research results for a specific case, but would not necessarily mean that our research is generalizable, and would not bring supportive evidence that the framework can be used in various other agile project implementations. This brings us to the second future research question that we would propose.

(2) What metrics, measures and measurement practices are applicable to a wide range of diverse agile software development projects for measuring project value?

As our research was based on theory and validated based on the empirical data collected from a single firm, the generalizability of our findings does not have strong evidence for practical use. Because one of the strongest blockers of implementing agile value measurement is the lack of resources and expertise that know how to implement, conduct and interpret value measurements, perhaps it would be interesting to understand what could be an effective baseline from where to start such a value measurement program. Just as with other types of IT management frameworks such as COBIT or ITIL, the framework would need to contain best practices and suggest the governance structures necessary for any agile project, regardless of specifics. The results of such a research would greatly help IT project managers and owners to bring measurement into focus, and to not be discouraged by the costs of "reinventing the wheel" for each specific project that requires measurement. We believe that to obtain meaningful results, a quantitative study would have to be conducted to review a large number of companies utilizing value measurement of their IT projects, especially focusing on highly experienced professionals that understand how measurement is done and how it is governed. The data obtained would then be analyzed to understand which metrics, measures and practices are common across a the most number of projects (keeping in mind that these projects should also be different in their scope and objectives).

REFERENCES

Ambler, Scott W., and Mark Lines. Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise. IBM Press, 2012.

Banker, Rajiv D., et al. "Automating output size and reuse metrics in a repository-based computer-aided software engineering (CASE) environment." *IEEE Transactions on Software Engineering* 20.3 (1994): 169-187.

Benefield, Gabrielle. "Rolling out agile in a large enterprise." *Hawaii international conference on system sciences, proceedings of the 41st annual*. IEEE, 2008

Biffl, Stefan, et al., eds. Value-based software engineering. Springer Science & Business Media, 2006.
Boehm, Barry W., and Rony Ross. "Theory-W software project management principles and examples." IEEE Transactions on Software Engineering 15.7 (1989): 902-916.

Boehm, Barry, and Richard Turner. "Management challenges to implementing agile processes in traditional development organizations." *IEEE software* 22.5 (2005): 30-39.

Dale, C. J., and H. Van Der Zee. "Software productivity metrics: who needs them?." *Information and Software Technology* 34.11 (1992): 731-738.

De Wit, Anton. "Measurement of project success." International journal of project management 6.3 (1988): 164-170.

Dingsøyr, Torgeir, et al. "A decade of agile methodologies: Towards explaining agile software development." Journal of Systems and Software85.6 (2012): 1213-1221.

Elssamadisy, Amr. "Has Safe Cracked The Large Agile Adoption Nut? ". InfoQ. N. p., 2016. Web. 17 Nov. 2016.

Favaro, John. “Extreme Programming and Agile Processes in Software Engineering”
Volume 2675 of the series Lecture Notes in Computer Science pp 16-25 (2003)

Fehlmann, Thomas Michael, and Eberhard Kranich. "Early Software Project Estimation the Six Sigma Way." *International Conference on Agile Software Development*. Springer International Publishing, 2014.

Fenton, Norman. "Software measurement: A necessary scientific basis." *IEEE Transactions on software engineering* 20.3 (1994): 199-206.

Fowler, Martin, and Jim Highsmith. "The agile manifesto." Software Development 9.8 (2001): 28-35.

Gartner, Nondisclosed presentation

Hall, Susan; “IT/Business Hybrids on the Rise”, ITBusinessEdge.com article, January, 2011

Hartmann, Deborah, and Robin Dymond. "Appropriate agile measurement: using metrics and diagnostics to deliver business value." Agile Conference, 2006. IEEE, 2006.

Herzog, Christian, et al. "Methods and metrics for measuring the success of Enterprise Social Software–what we can learn from practice and vice versa." *21st ECIS* (2013): 1-12.

Heusser, Matthew. “Introducing the scaled agile framework”. CIO.com article, June, 2015.

Hevner, Alan R. "A three cycle view of design science research." Scandinavian journal of information systems 19.2 (2007): 4.

Highsmith, Jim, and Alistair Cockburn. "Agile software development: The business of innovation." Computer 34.9 (2001): 120-127.

Javdani, Taghi, et al. "On the current measurement practices in agile software development." arXiv preprint arXiv:1301.5964 (2013).

Jones, Capers. Applied software measurement: global analysis of productivity and quality. McGraw-Hill Education Group, 2008.

Kaner, Cem, Walter P. Bond, and Pat McGee. "High volume test automation." International Conference on Software Testing Analysis & Review, Orlando, Florida. 2004.

Kulas, Hanna. "Product metrics in agile software development." (2012).

Laanti, M.: Agile Methods in Large-Scale Software Development Organizations. Applicability and model for adoption. Dissertation. University of Oulu (2013) ISBN 978- 952-62-0033-0

Laanti, Maarit. "Implementing program model with agile principles in a large software development organization." Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International. IEEE, 2008.

Leffingwell, Dean, “Scaled Agile Framework (SAFe)”, http://www.scaledagileframework.com/, (2011)

Luftman, Jerry. "Assessing business-IT alignment maturity." Strategies for information technology governance 4 (2004): 99.

Marciuska, Sarunas, Cigdem Gencel, and Pekka Abrahamsson. "Feature usage as a value indicator for decision making." *Software Engineering Conference (ASWEC), 2014 23rd Australian*. IEEE, 2014.

Mahnic, V., and Ivan Vrana. "Using stakeholder driven process performance measurement for monitoring the performance of a Scrum-based software development process." *Elektrotehniski vestnik* 74.5 (2007): 241-247.

Morasca, Sandro. "Software measurement." Handbook of Software Engineering and Knowledge Engineering 1 (2001): 239-276.

Nerur, Sridhar, RadhaKanta Mahapatra, and George Mangalaraj. "Challenges of migrating to agile methodologies." *Communications of the ACM* 48.5 (2005): 72-78.

Olszewska, Marta, et al. "Quantitatively measuring a large-scale agile transformation." *Journal of Systems and Software* 117 (2016): 258-273.

Overby, Eric, Anandhi Bharadwaj, and V. Sambamurthy. "Enterprise agility and the enabling role of information technology." European Journal of Information Systems 15.2 (2006): 120-131.

Petersen, Kai, and Claes Wohlin. "The effect of moving from a plan-driven to an incremental software development approach with agile practices." *Empirical Software Engineering* 15.6 (2010): 654-693.

Peterson, Ryan R., Ramon O'Callaghan, and Pieter Ribbers. "Information technology governance by design: investigating hybrid configurations and integration mechanisms." Proceedings of the twenty first international conference on Information systems. Association for Information Systems, 2000.

Qu, Wen Guang, Wonseok Oh, and Alain Pinsonneault. "The strategic value of IT insourcing: an IT-enabled business process perspective." The Journal of Strategic Information Systems 19.2 (2010): 96-108.

Sekaran, Uma. Research methods for business: A skill building approach. John Wiley & Sons, 2006.

Shacklett, Mary. Tech Decision Maker, TechRepublic.com. “What’s behind enterprise insourcing of IT?”, November, 2015.

Sommerville, Ian. “Software Engineering, 9th Edition.” Pearson (2010)

Stellman, Andrew, and Jennifer Greene. Applied software project management. " O'Reilly Media, Inc.", 2005.

Tarhan, Ayca, and Seda Gunes Yilmaz. "Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process." Information and Software Technology 56.5 (2014): 477-494.

TechTarget.com, Goulart, Karen; “IT insourcing uptick: Survey shows some IT coming back in-house”. April, 2013

Verschuren, Piet, and Rob Hartog. "Evaluation in design-oriented research." *Quality & Quantity* 39.6 (2005): 733-762.

Vickers, Paul, and Camden Street. "An Introduction to Function Point Analysis." School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, UK (2001).

Vijayasarathy, Leo, and Dan Turk. "Drivers of agile software development use: Dialectic interplay between benefits and hindrances." Information and Software Technology 54.2 (2012): 137-148.

Zuse, Horst. A framework of software measurement. Walter de Gruyter, 1998.

Chapter 8 APPENDIX

8.1 Interview protocol - Interview protocol - Effectiveness and Business Value Measurement Framework

1. Introduction

Explain the nature of the study to the respondent, telling how or through whom he came to be selected:

1.1. Goal of the study:

Objective: To deliver a tool in the form of a framework that should be utilized by IT project stakeholders that are responsible for measuring project efficiency, effectiveness and value.

Context: Measuring the value, cost, quality and efficiency of IT project deliverables is a challenging and often discounted part of organizational IT management, especially when it comes to measuring effectiveness of the solution with respect to business needs. Traditionally, the measurement of IT project value and performance has been approached from a technical, efficiency-oriented perspective (Jones, 2008).

Efficiency oriented measurements measure performance of software, project costs, productivity and other factors (Jones, 2008; Zuse, 1998;), which provide insights into the resource costs (input) of projects, but are weaker in reflecting business value (output). With the increasing presence and influence of IT in business, a higher degree of integration between the two is needed for IT to deliver value.

From a performance measurement perspective, the increased **Business-IT hybridization** of projects calls for broader, more comprehensive practices for **measuring effectiveness**. Ideally, effectiveness measurements should reflect the value of delivered projects through metrics such as adoption rate, utilization, customer satisfaction, etc.

This approach can be extremely **valuable for decision makers** to correctly allocate resources, and manage increasingly complex and volatile IT portfolios. However, the ideal case is underdeveloped in practice - performance measurements are mostly efficiency and technical-oriented, meaning that business stakeholders rarely receive any metrics relevant to them (CEB).

In the **case of Shell**, business and IT managers feel the need to complement existing performance measurement methods with value measurement, and utilize a consistent method across different projects and teams within the IT function.

Therefore, **the problem** can be phrased as: There is no consistent tool or methodology to measure the effectiveness and business value of hybrid IT-business projects and portfolios within the Scaled Agile framework.

1.2. Benefit for the interviewee: Interview is the basis for further improving the different metrics and measurements considering the different views of people within the organization, gives interviewee the

chance to contribute to the improvement of the measurement framework that they might apply in the future

1.3. Summary/Contents of the interview:

- Introductions and experience
- Rating existing software product KPIs
- Suggesting new KPIs
- Suggesting recommendations for applying the framework
- Closing remarks
- **Time:** 1 hour

Give assurance that respondent will remain anonymous in any written reports growing out of the study, and that his responses will be treated with strictest confidence.

Indicate that he may find some of the questions far-fetched, silly or difficult to answer, for the reason that questions that are appropriate for one person are not always appropriate for another. Since there are no right or wrong answers, he is not to worry about these but to do as best he can with them. We are only interested in his opinions and personal experiences.

Interviewee is to feel perfectly free to interrupt, ask clarification of the interviewer, criticize a line of questioning etc.

Ask if the goals and high-level contents of the interview are clear and relevant for the respondent.

Interviewer is to ask permission to tape record the interview, explaining why he wishes to do this.

2. Experience

- What is your professional background (how long at the company, education)?
- On which project/product are you working within Shell?
- What is your role within the life-cycle of the project/product (short description)? Include information such as department, discipline (there are a number of pre-defined disciplines at the company for different development activities).
- How long have you been working in this role?
- In which other disciplines have you been working and for how long?
- What is your experience with traditional and agile development? Select from the following options with multiple selections being possible (has to be done once for each model):

Traditional development	EDGE/Agile/Scaled Agile development
No previous experience Studied documentation	No previous experience Studied documentation

Traditional development	EDGE/Agile/Scaled Agile development
Informal discussion with colleagues Seminar and group discussions Used in one project (started or completed) Used in several projects	Informal discussion with colleagues Seminar and group discussions Used in one project (started or completed) Used in several projects

3. Main Body of the Interview

3.1. Existing measures

- Which of the following categories of KPIs and measures have you/your team previously used for measuring effectiveness and business value of software projects?

KPI/Metric/Indicator	Sub-indicator	Measure	Uptake/Use/Satisfaction
Fulfilment of market needs		Change requests/requirements	Satisfaction
Quality	Snag metric	Total no. of external reports - users	Satisfaction
Customer/business satisfaction	quality of product, reliability of time and costs, completeness of product delivered at the end of each Sprint or release, flexible handling of changes	Survey results	Satisfaction
Customer needs/value	Big data usage	helpdesk tickets, feedback from support interventions, knowledge management and forums	Satisfaction
Indicator Development for Decision Making Process	Usage threshold	relative actual usage by the participants, the intended usage by the developers when developing	Usage

KPI/Metric/Indicator	Sub-indicator	Measure	Uptake/Use/Satisfaction
		these features and the actual usage threshold indicating candidate features to be removed/improved	
Usage Value (demonstrates value for user)	Content & Usage Analysis	Adjusted Ideas (number of ideas about features, functionality and software)	Satisfaction
		Intensity of collaboration (number of users interactions with each other based on software content or functionality)	Uptake
		Degree of cross-linking (range of users linked together)	Uptake
	Database / Log File queries - for Process compliance as well	Log-in number and frequency	Usage
		Feature number of times used, frequency, duration	Usage
		New users gain rate OR unique visitors	Uptake
		Hits per time period	Usage
		Number of sessions, duration, average time per visit, frequency	Usage
	User interviews and surveys	User requirements	Satisfaction
		Usage behavior (sequence of steps, inputs, etc)	Usage

KPI/Metric/Indicator	Sub-indicator	Measure	Uptake/Use/Satisfaction
		Use case validation	Usage
		User satisfaction with tools and processes	Satisfaction
		Applicability of tools / knowledge of users about their applicability	Usage
Process Compliance	Database/Logs	Is the sequence of steps as the one intended by the development team/business?	Uptake
	User Interviews & Surveys	Effort of utilizing the software/tools	Satisfaction
	Reusability	Was the application developed to meet one or many user's needs?	Uptake
	Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?	Uptake

- Which of those metrics directly (or in combination with others) have you confirmed to influence the business metrics (financial, market)?
- How did you go about measuring these metrics?
- How did you use the outcome of the measurement? For example: did you share them with business stakeholders, IT management? Were decisions taken based on the results of the measures?
- Were these measures only relevant to your specific team/area or can they be applied to IT in general?
- Out of the remaining measures which of them might also influence the business metrics?
- Are they measurable and how would you go about measuring them?

3.2. New measures

- What other measures/KPIs besides the ones mentioned above would you suggest for measuring software that influence business metrics and are measurable?
- How would you go about measuring these metrics?
- Do you have any suggestions on how such a measuring framework could be applied in an EDGE/agile environment?
- Is it any different than in a waterfall project?

  4. Closing
- Is there anything else you would like to add that you think is interesting in this context, but not covered by the questions asked?

8.2 Coding of interview responses onto categories

Respondent 1 - HR IT portfolio Manager		
Data Categories	Data Units (Subcategories)	Coded (reduced) response
Context Data	Professional background and role description	Over fifteen years experience with HR Information systems, understanding of applications and the processes they support, and managing application lifecycles. The respondent's role is situated at HR portfolio and project level, where she collaborates with business analysts and project managers to ensure HR project goals are delivered to business stakeholders.
	Agile / Waterfall experience	Respondent has knowledge of agile methodologies and practices in IT. Agile delivery has been recently introduced for projects she works on, and plan-driven methods are still dominant.
	Project description	HR Strategy data integration project within 40 different applications. Applications have both internal (Shell) and external clients (partners for HR function).
Metrics Recognition	Satisfaction metrics used	Only the number of tickets measure has been tracked but not analyzed or used for decision makin. Barely any of the listed metrics have been used for measuring value in the respondent's projects due to the focus on meeting deadlines rather than quality or usage.
	Uptake metrics used	
	Usage metrics used	
	Novel metrics proposed	No novel metrics added, but respondent emphasized that the categories of end-user satisfaction and uptake should be

		monitored and analyzed for project success.
Metrics application	Metrics that influence business/financial measurement	User satisfaction is the most important for influencing financial measures. User login metrics should be used for decision making on discontinuing or keeping applications live (post-release).
	Decision making based on metrics	In this case (and many other cases of IT deliver), the business side has the implicit responsibility of making sure that an application has acceptable uptake and usage. IT Delivery is charged to include tools for monitoring metrics so they can be interpreted by business. Tools do exist in the IT applications delivered. However, these tools are not used.
	Agile vs. Waterfall measurement	The transition to agile development for some parts of Shell's IT is an excellent window of opportunity to introduce measurement projects. Measurement should be added across all IT portfolios, and not just for Agile, for measuring the value of projects
	Challenges and solutions of measurement	Respondent feels that measurement is hard to introduce because of Shell's traditional way of development (that did not include value measurement, and end-user related measurement). She reports that introducing a framework is mostly a political and change related problem, and believes that the measures and metrics exposed should be applied in any type of application and project.
Respondent 2 - IT Solutions Architect		
Data Categories	Data Units (Subcategories)	Coded (reduced) response
Context Data	Professional background and role description	IT consultant for the past 24 years.
	Agile / Waterfall experience	Most of his career, the respondent has worked with traditional development methodologies, but has reoriented towards agile development in the past four years.
	Project description	Application development projects for internal clients and functions.

Metrics Recognition	Satisfaction metrics used	Delivery teams do not use satisfaction metrics because of project costs. Because these time and money constraints, satisfaction should be predicted upfront by embedded controls (eg. sharing mock-ups with customers). Respondent reports that customer satisfaction should not be measured at the end but continuously tested during development and delivery, even for non-agile projects.
	Uptake metrics used	The business side (not delivery) utilizes usage and uptake metrics but these are not standardized or communicated to delivery and collaborated upon for decision making. As a result, there are no change requests towards delivery based on usage data. Functionality for measuring the metrics mentioned exists embedded in the software and by usage of tools, but is not easily reported on by business. A standardization of usage metrics is advisable.
	Usage metrics used	
	Novel metrics proposed	Business should decide what metrics and goals should be followed depending on case, and IT delivery should handle business request for implementing measurement and reporting tools, including telling the business how much it would cost. Metrics that are based on big data are very useful but very difficult to match with actual problems so that decisions can be relevant to create solutions
Metrics application	Metrics that influence business/financial measurement	Big data-based metrics on usage and uptake are essential for detecting spikes and identifying problems which are often not related to the technical functioning of the software, but rather to the learning curve of people using the software.
	Decision making based on metrics	Measurement is rarely made based on the metrics exposed in this research because the company focuses on business requirements, and not on actual usage. This is also because the company is not a customer-satisfaction oriented type of business, but focuses on utility.
	Agile vs. Waterfall measurement	Agile methodologies and these types of metrics mutually fit each other, because such measurement requires high collaboration and dynamicity. With Agile it is much more natural to include value measurement during development, because metrics can be assessed at the end of each sprint instead of the end of a whole

		project. The costs of agile projects become higher but there is less rework that has to be done due to sprints that catch issues in the development process.
	Challenges and solutions of measurement	In general, it is difficult to draw conclusions based on using measurement tools because there can be no mechanisms embedded in software to judge the context in which a measurement was made. It remains for people to make metrics usable and create a common understanding of KPIs and rulesets that can compose meaningful metrics. Project success (and measurement success) is highly dependant on skill, interpersonal relationships and clear goals understood by most parties involved. Making a measurement framework too specific (for clearer results and decision making) determines much higher costs and loss of efficiency in delivery of software because resources need to be dedicated for managing the measurement framework.
Respondent 3 - IT product Owner		
Data Categories	Data Units (Subcategories)	Coded (reduced) response
Context Data	Professional background and role description	Developer, business analyst, release planner and presently product owner of IT web project. Within the present role he is the chief responsible for delivery of the IT product (web applicaition).
	Agile / Waterfall experience	Full experience of waterfall development methods, and recently with 4-5 years experience with working in Agile. Continuous process of adapting work towards agile methodologies.
	Project description	The project is mostly external, with requirements developed by local business stakeholders for a client-facing web application. Development is done in a distributed manner (several locations off-site).
Metrics Recognition	Satisfaction metrics used	Most metrics recognized but non of them used. Satisfaction is established thorugh feedback discussions among project stakeholders and product owner. Metric collection mechanisms exist embedded in processes and tools but they are not used for decision making. New feature development or improvement are

		done based on requests from business but not by direct contact with end users or measurement.
	Uptake metrics used	Sequence of user steps, time of visit, logins, etc. is technically possible but not analyzed by business stakeholders.
	Usage metrics used	
	Novel metrics proposed	AB Testing as a direct feedback measure on deciding between two or more features, function or user experience choices by analyzing the responses of users with between these choices.
Metrics application	Metrics that influence business/financial measurement	Currently there is no metric used in this way, but business stakeholders should define a product's success with respect to a joint measurement of financial and value metrics. Respondent argues that the link between business (financial) and satisfaction of end-users is difficult to establish.
	Decision making based on metrics	Currently no decision making done based on metrics, but the business side should push initiatives for data analysis and decision making based on value metrics. These initiatives are highly dependant on communication between business and IT development stakeholders so that data can be analyzed and decisions taken.
	Agile vs. Waterfall measurement	With waterfall, the value metrics and measures specified cannot be applied during development because there is too little interaction with clients, and neither are they applicable at the end of the development lifecycle because it would prolong the cycle too much. Agile is thus the suitable work methodology for incorporating satisfaction, usage and uptake measurements throught development and after release.
	Challenges and solutions of measurement	Measuring the satisfaction of clients is difficult because there are wide differences between the degrees or definitions of satisfaction. Moreover, even if data is obtained, it is challenging to analyze it due to the lack of a methodology and structure for analysis. The solution is to include measurement processes and responsibilities within a project development lifecycle and in the interactions between the delivery and business stakeholders.