



Agile in action

Hybrid methodologies in practice

Emil Burman

Abstract

In recent years an increase of agile methodologies and their implementation in mainstream software engineering projects have been seen, but this implementation has reportedly changed agility in practice into a hybrid form of agility and more classic methodologies. With a lack of research regarding these hybrid methodologies this multiple case study empirically analyses the practice of methodologies from two different conglomerates and places the results in the Information Technology Management framework (ITMF) to answer the research question; How are hybrid agile methodologies enacted in practice in mature agile teams and why are they enacted in such a way? The result show that the organizations work agile within a waterfall framework, a hybrid form of requirement handling into agile development followed by waterfall testing. The hybrid methodologies are imposed by external factors such as the sales department and traditional economic reporting. This study acknowledges a need for a greater understanding of the engagement between external actors and software development.

Keywords: Hybrid methodology, agile methodology, project management practice

1. Introduction

Software engineering projects have a reputation of having a high risk of failure with \$80- 140 billion per year being spent on failed or cancelled projects (Hass, 2007). This is a trend that is tied to the project form, with some even predicting failure before the project starts and project failure rate ranging between 11% and 15% (Emam and Koru, 2008). Historically this risk has been controlled through the use of software engineering methodologies to establish a relationship between the project result and the project process (Nisa and Qureshi, 2012). This technique of using a methodology has been prevalent since the late 1960s (Avison and Fitzgerald, 2006) and traditionally has been governed through the waterfall methodology, whereas once a phase is completed it is assumed the project will not return to it (Hass, 2007). With the introduction of the agile manifesto (agilemanifesto.org, n.d) in 2001 another form of project philosophy was presented in the form of agile methodologies like Scrum, XP and DSDM where Scrum is by far the most popular. This agile philosophy was focused on human interaction, communication, customer interaction through deliverables containing software and an incremental process where the project can revisit earlier phases (Vinekar et al, 2006). In the years following the introduction of agile methodologies several studies have been published focusing on the implementation of agile methodologies in traditional project environments (Winter et al, 2006) but few studies have since been empirically focused around mature agile teams which have been using agile methods such as Scrum (Ahlemann et al, 2013) for a longer time and their implications within the software engineering industry (Dybå and Dingsøyr, 2008). The extensive use of Scrum in project methodologies is supported by West (2011) which study shows that agile is the preferred methodology for IT professionals with Scrum being the most common methodology used within agile

methodologies. West (2010) highlighted that although adoption of agile methodologies is high, agility is changed in practice and in actuality teams may adopt certain agile techniques that fit into their existing organization but not necessarily adopt the agile philosophy as a whole. Due to many studies being centered on implementation and not on the practice of agile methodologies after the initial adaptation it exist few empirical studies of agility based on the practice of agility (Dybå and Dingsøyr, 2008), creating a discrepancy between theory and practice (Conboy,2009). Thus while there exists plenty of theoretically created models regarding agile and hybrid methodologies few have actually analyzed the practice of agility (Cicmail et al, 2006). The change of agility in practice into hybrid methodologies is thus under analyzed and in need of further exploration.

1.1 Purpose and research question

With agility being changed in practice and with a lack of empirically focused research beyond the point of adoption of agility there is a concrete lack of research regarding the change of agility in practice in mature agile teams and how they work within a hybrid methodology. This study will thus answer the research question; *How are hybrid agile methodologies enacted in practice in mature agile teams and why are they enacted in such a way?*

1.2 Layout of the thesis

Section two will initially give a background of methods and methodologies and how hybrid methodologies fit within that background in order to differentiate the practice of hybrid methodologies from both agility and the waterfall model.

Section three continues by presenting earlier research and how this study will be aligned to previous work.

In section four the Information technology management framework (ITMF) is presented and described to give an overview of its parts before moving on to section five.

Section five gives an overview of the method and the research approach. It motivates both the design and conduct of the study as well as the considerations made to maintain anonymity for the respondents.

Section six presents the result by dividing the data into two sub sections with each section pertaining to one part of the research question, thus giving structure to both how things are enacted in practice and why it is enacted in such a way before moving on to the analysis.

Section seven places the data into the ITMF to illustrate how the application of hybrid methodologies are enacted in practice in certain stages. The model can be used to compare both agile and waterfall influences in the form of work practices and thus allows a greater understanding of their interaction.

Section eight and nine present discussion and conclusion of the study and highlight areas for future research.

2. A historical view of development methods and methodologies

This section will give a quick overview of the historical origins and differences between the concept of methods and methodologies as well as the difference between three schools of software development methodologies.

2.1 The concept of methods and methodologies

The methodology and project management field is not an emergent field in itself, on the contrary it is quite established. Despite this the field has traditionally had issues connecting theory with practice, with numerous troubles concerning methodologies (Abrahamsson et al, 2009). Some researchers showcases that the lack of adoption of methodologies stem from a lack of empirically and theoretically connected research (Ahlemann et al, 2013) and others state a lack of customization of methodologies is to blame (Ward and Legorreta, 2009). Although methodology as a concept is widely used it lacks a universal definition, which Avison and Fitzgerald (2006) highlights with a general definition of a methodology as simply a recommended series of steps in a software engineering project, a definition that does not clearly separate it from a method. Questions such as if a collection of tools constitutes a methodology or if the use of a methodology need to produce the same result every time arise with such a loose definition. Thus it is needed to define what the difference between a method and a methodology is, as well as what separates the different methodologies before defining hybrid methodologies.

Methodologies are used for different reasons and as such have a variety of forms, Avison and Fitzgerald argue through the use of Maddison (1983) that a methodology is; “*a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems.*” (Avison and Fitzgerald, 2006)

They continue to add several concepts to further define a methodology concerning what tasks, outputs, constraints and resources should be produced and carried out at certain stages of the project. To further distinguish a methodology from a method they bring forward the concept of philosophy and how it is incorporated in a methodology, making it broader than a single method. Thus a method could be contained within a methodology, since it does not have an overarching philosophy behind it and thus can be used as a collection of techniques resembling the use of a tool. Finally Avison and Fitzgerald define a methodology as;

A systems development methodology is a recommended means to achieve the development, or part of the development, of information systems based on a set of rationales and an underlying philosophy that support, justifies and makes coherent such a recommendation for a particular context. The recommended means usually includes the identification of phases, procedures, tasks, rules techniques, guidelines, documentation and tools. They might also include recommendations concerning the management and organization of the

approach and the identification and training of the participants. (Avison and Fitzgerald, 2006 p. 568)

With the concept of a methodology established there is a need to further distinguish what defines the philosophies of traditional and agile software development as well as what distinguishes them. Next I will briefly discuss the three strands of project management schools (Winter et al, 2006) and how hybrid methodologies fit within them.

2.2 Chronological evolution of methodologies

Formalized software engineering methodologies emerged between the beginning of the 1970s (Avison and Fitzgerald, 2006) and the late 1960s (Winter et al, 2006) and focused on the identification of phases and stages to help control the management of a project. The biggest methodology was named the System Development Life Cycle, or SDLC for short. Later it became commonly known as the waterfall model (Avison and Fitzgerald, 2006). Its sequential execution and use of documentation standards coupled with phases divided into manageable tasks has given it a 'true and tested' staple. Of the three strands of project management schools this is deemed the 'hard strand', which focuses on control and planning of projects (Winter et al, 2006). Even if there were several alterations and deviances of the SDLC they all usually shared control stages which all had clear handovers between them; feasibility, study, systems investigation, analysis, design, development, implementation and maintenance.

The waterfall model was not without critics; with its focus on control it lacked flexibility and an inability to adapt to emergent front end work. (Winter et al, 2006) As such in its wake several other methodologies began to form from two different sources; theory or practice, which in turned formed the second strand of project management schools. These focused on organizational structure as a mean to achieve task completion and integration (Winter et al, 2006). Today the methodologies developed from practice are most widely known and while most early methodologies were created ad hoc and commonly relied on one technique or strongly coupled techniques such as entity modeling or data flow diagramming, they slowly turned into fully fledged methodologies (Avison and Fitzgerald, 2006). A combination of techniques created blended methodologies such as SSADM and combined with an expanded scope for methodologies as a concept as well as a demand to fill gaps in their process the concept of methodologies expanded.

With the expansion of methodologies they became more complex and in the mid to late 1990s many organisations saw their adoption of a methodology fail. This in turn saw heavy criticism of the methodology concept as a whole and complaints of heavy complexity, one-dimensionality as well as difficulties adopting a methodology among many other issues began to arise (Avison and Fitzgerald, 2006) which created the third strand of project management.

The third strand of project management focused on context, communication, learning, strategy and how project experience is crucial to project management (Winter et al, 2006). This strand released a manifesto in 2001 named 'The agile manifesto' (Agilemanifesto.org, n.d.) which focused on customer satisfaction and interaction, welcoming of changing requirements and integration and communication of project and business actors. Continuous testing alongside developers coupled with team rooms to increase communication are staples

of agility. Refactoring, returning to written code to find common elements that can be generalized, as well as continuous integration, automated testing at different levels, are both prominent factors of different agile methodologies. With this being the latest strand of the project management school it still lacks a generally accepted definition, as such I need to be more specific in relation to agility and the agile concept.

2.3 Agility

The concept of agility has commonly been poorly defined (Dingsøyr et al, 2012) and therefore I will in this study rely on the work of Conboy (2009) as well the work of Christopher and Towill (2011) to define agility as well to further distinguish agility from the concept of leanness and flexibility. It is important to make a distinction between agility in other literature and agility in relation to software engineering project methodologies. For instance Christopher and Towill (2011) define agility in relation to supply chains as: *“Agility means using market knowledge and a virtual corporation to exploit profitable opportunities in a volatile marketplace.”*

While the agile methodologies in a high abstract view fits within this broad definition there is a need for increased accuracy and contain it within information software development (ISD). This study rather relies on the work of Conboy who makes a point to distinguish between flexibility and agility, two concepts which at times have been used interchangeably. Conboy (2009) defines flexibility as: *“...the ability of an ISD method to create change, or proactively, reactively, or inherently embrace change in a timely manner, through its internal components and relationships with its environment.”*

To further define agility it is important to differentiate between agility and leanness. While value is represented in leanness it lacks the focus on customer driven value that is central to agility. It also can not handle variability to the same extent as agility, instead it is better suited for things you control (Conboy, 2009). Christopher and Towill (2011) define the concept of leanness accordingly: *“Leanness means developing a value stream to eliminate all waste including time, and to enable a level schedule.”*

While leanness have a similarity to agility and are in certain ways interconnected Conboy (2009) argue that agility is better suited for a learning organization where flexibility and variability is needed. To make this clearer there are other concepts that are contained within agility which differentiates it further from flexibility and leanness. Concepts such as speed, exploration as well as continuous change are all part of the agile concept. As such Conboy defines agility accordingly, which will be the definition of agile and agility in this study:

*The continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.
(Conboy, 2009 p. 340)*

2.4 Hybrid methodologies

The agile methodology has taken the software engineering world by storm; some even stating its processes are often necessary for a successful project (Björkholm, Brattberg, 2010). But the reality of agile adoption is not complete adaptation of the methodology but rather a hybrid form with areas that remain within other methodological philosophies (West, 2010). The emergence of hybrid methodologies share similarities to the emergence of blended methodologies, for instance SSADM that combined two schools of thinking into a single methodology. (Avison and Fitzgerald, 2006) A phenomenon that could be repeating itself with the adoption and use of agile methodologies. A hybrid methodology is the combination of one or several agile methodologies coupled with other strands of project management schools. With a lack of studies within the hybrid field there does not exist a clear definition of the concept similar to that of agility stated above. Due to this there is a lacking understanding of a hybrid philosophy since few studies have been performed on the underlying motives of hybrid methodologies. In relation to this thesis the concept of hybrid methodologies will be defined as the mixture of two or more different methodologies from two different schools. For instance the hybrid methodology created in a controlled case study by Nisa and Qureshi (2012), which successfully combined Scrum, XP and RUP into SPRUP. While there is a lack of studies regarding the reasoning for using hybrid methodologies there are issues with adopting agility, which might be a reason for using hybrid methodologies.

In order to fully differentiate between agile and hybrid methodologies current research will be presented in the next section, moving away from the historical perspective into a current one.

3. Current research within development methodologies

This section is confined to research related to agile and hybrid methodologies and the recent evolution of those sub-fields within project management. It moves past the historical perspective and briefly touches upon agile research before moving on to current research regarding hybrid methodologies.

3.1 Agile and hybrid methodologies

Agile methodologies have had research nearly to the point of saturation on the adoption of agility (Boehm, 2002; Nerur et al, 2005; Boehm and Turner, 2005; Nerur and Balijepally, 2007), how adoption of agility into an organization should be balanced in regard to traditional methodologies (Boehm and Turner, 2004), discussing agility as a toolbox (Cannizzo, Marcionetti and Moser, 2008), how to facilitate communication and trust to create cross functional teams (Moe et al, 2009; McHugh et al, 2012) as well as the effectiveness of continuous testing (Erdogmus et al, 2005; Stolberg, 2009), the latter since continuous testing is a staple of agility. Of course there exists research regarding what some view as the negative sides of agility, for instance myths regarding no documentation being required, a higher quality product as a result of using agility as well as agility being the

answer to changing requirements (Hochmüller and Mittermeir, 2008). Furthermore other research highlight the issues with communication with the client (Krasteva and Ilieva, 2008) and teams feeling they do not have the freedom agility should give them.

In comparison to the myths the critical success factors regarding the use of agility have through a quantitative study been narrowed down to three factors; a correct delivery strategy, a proper practice of agile software engineering techniques and a high-caliber team (Chow and Cao, 2008). Other factors such as a good agile project management process, an agile-friendly team environment, and a strong customer involvement could possibly be critical but lack the same connection as the above factors according to Chow and Cao (2008).

The critical factors of agility are important in the agility versus waterfall discussion and through a unifying framework Glaiel, Moulton and Madnick (2013) analyses the cost, time and quality between the two methodological schools. They initially create a agile genome which categorizes agile aspects such as breaking down requirements, incremental development, micro-optimization (cross functional teams) and customer involvement through feature driven development. This genome is then incrementally inserted in the experiment to study its effects on the project. When the most common form of agility, which does not include refactoring and continuous integration, is compared to waterfall it saves some cost, improves quality and delays the project in the experiment by three weeks. Depending on the project this could either be viewed as positive or negative. The study continues to showcase agility as a toolbox, which can counterbalance project choices and can thus be negative or positive on the project explaining that the most common adoption of agility is through feature driven incremental development containing cross-functional teams. It should be noted that when fully adopting the agile genome the project experiment produced a shorter schedule, a slightly higher cost but also the highest quality product (Glaiel et al, 2013).

In regards to such agile research several critics of the research itself have emerged. Dybå and Dingsøyr (2008) conducted a review of several studies performed empirically within agile software development, a review that covered 36 relevant articles of the agile field. Besides simply analyzing the content of the papers they highlighted a need to increase both the quality and the amount of studies being performed. This notion is supported by several other researchers (Conboy, 2009 | Cicmail et al, 2006 | Abrahamsson et al, 2009) and a recent report by Falessi et al (2014) feature the emergent areas of agility. These areas highlight maturity within agility and agile teams as being integral and in need of more research.

Prior research regarding teams includes the challenges of using Scrum in distributed teams (Hossain et al, 2009) and how trust in globally distributed agile teams affect communication (Dorairaj et al., 2012). Jalali and Wohlin (2010) examined such distributed agile teams and highlighted a change of agility in practice, a general change supported by West (2010).

Through a literature study a solution to this change of agility in practice was proposed with an ambidextrous view of agility and the waterfall model by Vinekar et al (2006). They showcased that organizations see a need for both agility as well as waterfall and as such propose an ambidextrous hybrid solution of subunits. West (2011) on the other hand states

through a quantitative study that the current state of agility in organizations exists as a hybrid form of agility and not an ambidextrous one. With agility being a toolbox where actors can choose which methods they want to use, many mix and match from different methodologies to create their own methodology. The rarity of a single methodology being used is a common phenomenon (Lozo and Jovanović, 2012) and many researchers have created theoretical hybrid models to imitate practice. These models often use Scrum coupled with other methodologies (Mushtaq and Qureshi, 2012; Rasool et al, 2013; Castilla, 2014) but commonly lack the connection to practice that is sought after in this field. Hayata and Han (2011) takes this hybrid view and creates a hybrid model, combining waterfall at the beginning into Scrum during development and with a transition into waterfall testing at the end of the project. Their study is supposedly empirical but the article lacks information on how it was conducted and a general lack of sources to rely on, making the model within it mostly theoretical.

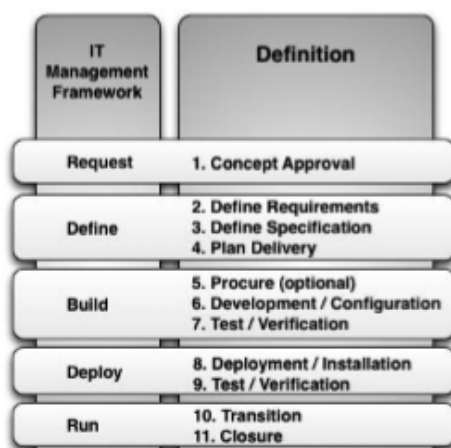
3.2 Positioning of the research question

In comparison to Nisa and Qureshi (2012), which created a methodology in a controlled case study, this study focuses on the application of agile methodologies in mature development teams *in action*. Mature teams in this study are specifically teams with a minimum of two years prior experience with Scrum and as such have moved beyond the point of simple adoption. In order to research which part of their methodology is agile and why there exist influences of other strands of project management schools within their methodology there is a need for the teams is mature in order to find reasons for the hybrid form beyond remnants of the adaptation period. This leads to my research question; *How are hybrid agile methodologies enacted in practice in mature agile teams and why are they enacted in such a way?*

4. Theoretical frameworks

Comparisons between methodologies are always difficult (Avison, Fitzgerald, 2006) and as such there is a need to find common ground to make the comparison possible. For this purpose the Information technology management framework (ITMF) is used. It is designed to place methodologies within common more abstract concepts, making it easier to understand where they differentiate.

4.1 Information technology management framework (ITMF)



The ITMF created by Pollard and Geisler (2014) is a framework created to allow an organization to gather information from multiple projects that are using different methodologies and frame it within a common framework lifecycle. Through analyzing agile, SDLC and other project management philosophies Pollard and Geisler created an overarching lifecycle framework to find common themes within several methodologies.

With methodologies being customized and even used in a hybrid form (West, 2010, 2011) there is a need to find a common way to compare them. ITMF allows comparisons between methodologies and how they are enacted in practice by bringing them under the same concepts, allowing information to be standardized. The framework as seen in Fig. 1 is a conceptual model of the ITMF, which Pollard and Geiler had a few methodological lifecycles in mind when creating. Methodologies such as Scrum, SDLC and experimental models all have their lifecycles represented and the overall structure of the ITMF have five sections that are defined accordingly:

1. **The request stage** defines the objectives and goals of the project and motivates the need for change.
2. **The define stage** refines the requirements and specification and plans the execution of the project together with the project stakeholders. This stage produces a common position between the stakeholders of the software engineering project and the delivery areas, which could be in several forms ranging from requirement documents to storyboard cards or a prototype.
3. **The build stage** is the construction of the delivery areas through procurement of resources needed for the project as well as the development and internal tests of the deliveries. In many methodologies this stage would go under the implementation concept.
4. **The deployment stage** is the integration of the deliverable into the business level where testing will be done within the environment where it will be eventually run. For instance this stage would fit within the testing stage and verification within the SDLC methodology.
5. **The run stage** is the transition from completed requirements by the project owners to business ownership of the deliverable. It also contains the support given after the project have delivered and closed its teams.

Due to the frameworks independence from any methodology it removes any ambiguity from difference in concepts and terms thus removing the apparent difficulties in comparing linear or cyclic methodologies. This will facilitate the comparison between the different interviews and the methodologies therein. When comparing the methodologies their defined stages will be placed in this framework in order to use the same concepts in the latter parts of this study.

5. Method

Within this section I will briefly discuss my data collection method and analysis. Here I will also give a short description of the use of the ITMF and its application as an analysis tool within the study as well as how the CHAPL framework was used as an inspiration for the data collection.

5.1 Research approach

The difference between qualitative and quantitative research methods is debated (Bryman, 2011) but in this study qualitative data collection method was chosen since qualitative

research methods focus on in-depth knowledge (Patton, 2002) and theory generation (Bryman, 2011). Qualitative methods focus on interpreting and understanding social realities within the subject's context (Bryman, 2011), which in turn aided the understanding of why the methodologies were chosen and enacted in a certain way. With the research questions centering on *how* and *why* hybrid methodologies are used the choice of qualitative data collection through a case study was further supported, since this form is suited for uncontrolled studies focusing on temporary events (Yin, 1994). With the changing of agility in practice (West, 2010) I deemed it necessary to use interviews in order to interpret how agility was enacted in a hybrid methodology. The use of interviews allowed a nuanced view of the hybrid methodology concept, something quantitative methods would not have been able to do. With the research questions being loosely based on the example questions provided by the CHAPL framework the interviews yielded very distinct answers.

5.2 Study context

With the purpose to analyze how hybrid methodologies are enacted in practice and why they are enacted in such a way empiricism is the logical school to follow. Empiricism represents a view that knowledge must be based on reality and theoretical concepts must resign itself to reality (Bryman, 2011). This connection to reality is sought after in the field of project management and methodologies, further motivating the choice of data collection as a multiple case study that is distinctly empirical (Yin, 1994). A case study can be performed outside of ethnography as well as during an intense period of time, further showcasing it being suited for this study due to time constraints. Having multiple sources for the study ensures construct validity and allows greater accuracy, avoiding the results being happenstance. In a multiple case study it is important that the different cases are either similar, termed literal replication (Yin, 1994), or contrasting with predictable reasons, termed theoretical replication (Yin, 1994).

The cases chosen for this study is motivated by their critical representation (Yin, 1994) of hybrid methodologies. In order to increase accuracy of the study four organizations within two conglomerates were studied and placed within the ITMF. Constructing the study in such a manner allows the sample a wider depth in correlation to the interview sample selection while still maintaining equivalence in both interviews and the study itself.

5.2.1 Semi-structured interviews

Semi-structured interviews is a concept that is very broad, it incorporates a more general approach to the questions asked. This general approach to a respondent often requires follow up questions in order to clarify aspects of the respondents answer (Bryman, 2011). This approach uses pre-written questions in order to maintain equivalence between interviews, but does not follow it with the same rigidity as structured interviews. Instead semi-structured interviews are more open and iterative in its nature (Bryman, 2011). This allows an increased flexibility in the interview while still focusing on themes predetermined by the researcher. With the research question established early in the study coupled with the use multiple cases, semi-structured interviews combines the rigidity and flexibility to attain equivalence.

The essential issues presented by the CHAPL framework inspired the interview questions. The CHAPL framework; **C**ontextual, **H**istorical, **A**nalysis, **P**henomenological and **L**inguistic

was initially designed to give software engineers an objective understanding of methodologies in order to create a best practice for a software project (Jiang, Eberlein, 2008). The framework is a base to create reasoning for choosing a methodology and tries to analyze in depth the factors that influence the use of a methodology. Created with the intend to facilitate choosing a methodology by understanding its social and philosophical background this framework was the inspiration for the interview questions, providing the rigidity needed for equivalence between cases as well as the flexibility to retain the semi-structured form.

5.2.2 Sample selection

The data sample was made from a convenience sample initially through several contacts in the industry, which in turn made the selection of respondents thus moving on into a snowball sampling (Bryman, 2011). Being informed of the enactment of their methodologies prior to researching the organizations it was deemed their cases being representative of hybrid methodologies in practice as well as having literal replication, further motivating the data collection. The initial convenience sample provided contacts outside of the first organization, which in turn lead to the sample containing four project managers, two Scrum masters and two developer level employees, all which volunteered after being contacted in two different conglomerates. As seen in Table 1 below two locations had three respondents while two other locations had one respondent. Within organization A and B all interviews were conducted in person while in organization C and D they were conducted over the phone. All interviews were recorded and later transcribed to allow data analysis at a later time.

Conglomerate 1 has offices in two major cities as well as an office in a minor one. They are an IT consultant firm with a contract to deliver a major IT system to a bank during the time of the study. During this project the majority of the office in a minor city were involved, roughly 40 people. This office is showcased in the form of organization A, as shown below in table 1. Conglomerate 2 is one the biggest IT consulting firms in Sweden with offices in most of the larger cities in Sweden as well as in Scandinavia. The respondents were from different organizations within the conglomerate but they all worked within projects that involved the majority of the employees within their respective organizations. The size of those organizations varied from 30 people up to 70 people. Organization B, C and D are all part of conglomerate 2 but all worked independently on different projects.

Because of the size of the different projects as well as the respondents' knowledge about their methodology I feel confident that the sample can accurately portray the work with a hybrid methodology in this multiple case study. Since there does not exist any rules for sample size (Patton, 2002) this is in turn a discussion of breadth versus depth. I've chosen to have more in-depth interviews with fewer people since I deemed their experience to hold enough information for my study. Considering my interviews in themselves were information rich (Patton, 2002) I can once again claim satisfaction with the sample. While two organizations had one respondent their answers provided enough depth to keep them in the study. While other respondents with responsibilities similar to organization A and B could have given another view of the projects in organization C and D the data collected was sufficient to establish a understanding of their respective project methodologies. The

respondents have been placed in table 1 according to their position and organization with abbreviations to keep their anonymity.

	Conglomerate 1	Conglomerate 2		
	Organisation A	Organisation B	Organisation C	Organisation D
Project manager	PM-A	PM-B	PM-C	PM-D
Scrum master	SM-A	SM-B		
Developer level employee	DL-A	DL-B		

Table 1; Table indicating the organization for each respondent

5.3 Data analysis

The data analysis was performed by several read-throughs of the data and highlighting data through color-coding of the transcriptions to collect data that pertains to respective sections of the ITMF. Thus all the data concerning the build stage for instance was collected in one document. These sections were later collected in one document and placed within the whole framework in order to give a structured view of the practice of the methodologies rather than just certain sections of it.

5.4 Ethical issues

The ethical issues of the study are related to the opinions and anonymity of the respondents. Within the context of the respondents being provided in a snowball sample (Bryman, 2011) each respondent's contribution to the study had to be as anonymous as possible, to protect their privacy and retain the status quo of their employment (Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning, 2002). Their participation was completely voluntary and the participants were informed prior to the interview that they at any point could refuse to answer any question. While in the context of the study there are no personal ethical issues to have in mind complete anonymity was a priority to protect the respondents. The ethical issue of validity versus avoiding harm (Miles, et al., 2014) is always present while performing a qualitative study.

5.5 Ecological validity and study construction critique

In a qualitative study it is important to distinguish between an individual's perceived reality and the actuality of the situation (Patton, 2002). Since many of the subjects act as representatives of other actors in their level of the organization, or as complete representatives for their organization, it is needed to use heavy analytical rigor in order to find their perception of the situation and the generally accepted perception. This could have been solved through the use of focused ethnography (Knoblauch, 2005), where observations and interviews are combined to study a social environment and the events within it. While observations is a strong tool for analyzing the practice within a environment it is not a

suitable tool for analyzing software engineering methodologies over a short period of time, as the confinements of this study would have demanded. In order to fully comprehend why certain events transpire within the different projects, knowledge of how the projects are structured is required. Interviews provided that insight to an equal or greater extent than observations would have. Observations are a strong tool for data collection and could have improved the data, however the deadline and geographical position of the cases made this data collection method impossible to perform. With projects being several months to several years long the use of observations would have been too short to see the actuality of the project methodologies, further showcasing the positive sides of interviews.

6. Results

In order to accurately present the results they will be divided into two separate categories with each category pertaining to a section of the research question.

6.1 Enactment in practice

The project managers made it clear in their interviews that the hybrid form is often agility within a greater waterfall framework, or as (PM-B) puts it;

I believe that the project is functioning in an agile way, but we can't change the big framework as much as we want. We have these requirements that we need to fulfill, and we have a deadline we need to stick to. I don't think it is wrong to have a clear framework. Although you need to have certain liberties within that framework to work with. (PM-B)

This overarching framework is a common denominator across the cases and starts with the requirements. Changing requirements is a central concept of agility and there are several different practices noted in the cases. While the requirements are usually set before the project start they remain within a high abstraction level, thus allowing interpretation from the developers. This phase is debated within the different organizations, with a developer level employee summarizing their organization's requirement handling as;

The big project compared with the messy teams is very waterfall based with handovers. Other teams are in the middle. In the refinement process developers are involved with product owners, interpreting requirements. The whole scale still exists within the organization. (DL-B)

Several claim being initially presented with the requirements (PM-B, PM-C, PM-D), and continuing with continuous interaction with the client to create user stories (PM-C); *"There is a difference now from the classical way, the requirements are on a high abstraction level and we break them down into user stories."*

The requirements are usually set before the project starts according to a developer level employee as well as a project manager;

Most is planned. You use the waterfall model until you have decided what you want in the project. They (the developers) have been involved earlier and developed suggestions for solutions. But that was three years ago. (DL-A)

We received a requirement specification in the beginning but the requirements were defined on a high abstraction level and we had the opportunity to work with questions that allowed for a dialog with the client. (PM-C)

A notion several project managers (PM-B, PM-C) independently supports. Another project manager highlights that a well-constructed requirement specification is still a necessity (PM-A); *“If you haven’t received a well constructed requirement specification from the client when you enter the project... I mean shit in, shit out. That’s the truth.”*

During the development all cases claim complete agility but some claim the team involvement is not there;

We say we use Scrum but I can tell you that we don’t. We have picked parts out of it and we are getting closer and closer towards the method that is Scrum, but we aren’t there yet. (DL-A)

While there exists a notion of discrepancy from what certain respondents view as agility and the organization view as agility all respondent mention sprints, burndown charts and other tools connected to agile development. It is worth noting that all respondents mentioned a change in the length of the sprints during development, both increasing and decreasing their sprint intervals. A project manager summarizes methodology during development (PM-A) in the following way; *“In the big projects we use daily standups, retrospectives, pre-planning and sprint planning. To a certain point in the smaller ones, but we skip the daily stand-ups and such then.”*

The cases follow the agile philosophies within development, although retaining a common clear testing phase after development. This is succinctly highlighted by a team member (DL-A) when asked about testing; *“During bigger projects we have testing both during and after development.”* Continuous testing is a part of all project methodologies with one exception, whose project manager (PM-D) explains how this was enacted in a finished project where testing was only performed at the end of the project;

In the beginning we started agile with no specification phase, no design phase and no information phase. You worked from a specification and you had iterations where you could demonstrate the functionality, creating a sort of hybrid of the project. In the end we still had a complete test phase of the project, meaning we couldn’t see the risks during the project. It wasn’t easy to change anything in the middle of the project, it was also problematic to change the project timeline. (PM-D)

The waterfall framework with handovers is therefore especially clear at the end of the projects, departing from agile philosophies with a clear testing phase.

6.2 Rationality behind the methodologies

External factors are the driving force behind the overarching waterfall framework with project managers admitting customer interaction is an issue (PM-A, PM-C); *“It is a bit of a problem, to get the (customer) interaction on the right level.”* (PM-A), a developer level employee (DL-B) admits the sales department is not educated in Scrum and as such give a distorted idea of the project process before development even begins;

Sales have not followed how we work in development. We need to create other contracts. Contracts that are more focused on following the agile way of working. Costs that are being focused on a monthly basis rather than a predetermined cost, a bit of product versus service (DL-B)

Furthermore a project manager points out that agile projects are still expected to fit within classic economic reporting;

Sure there are clients which work agile as we do but there is still the economical section which doesn't want to lose control. They aren't ready and still want a check every month. The system does not support the agile way of working and is still very much within the waterfall model. (PM-D)

These issues force the project to maintain an overarching waterfall structure even if the development in itself is close to pure agility. A project manager (PM-A) explain why Scrum is still used despite working within a waterfall structure; *“The biggest advantage I see is that you have a start and stop with everything (sprints). That you have deliverables during the process.”* a Scrum master continues this by stating; *“You can't plan for three months and execute that plan for three months, install at the client and wait for the disaster. That way of thinking is gone.”* (SM-B), others claim an increase in risk assessment being the reason for using agility during development;

You can see the risks faster. The downside is that you lack the long-term view. When questioned from people that develop the old way you can't really answer them. If they ask what we will do in 9 months we don't know. We have an end-game and a plan but we haven't considered later development, we will do that later in development. (PM-C)

A project manager (PM-B) continue to explain that this larger framework is a necessity in order to work with the client of the IT project, it gives participants a certain clarity regarding deliverables that wouldn't exist within a complete agile project. Another project manager separately continues this chain of thought by stating;

The agile way is a way to develop, to create. Waterfall is a way to plan a project, to plan the whole thing. Then there are ways to work within it. As project manager it is still the same work. Make sure they are communicating, solve issues that might appear. Deliverables are still the reality. They (the client) expect to get a long-term project plan to plan their own work. (PM-C)

In comparison another respondent answered the question in the following way;

We have a lot more mature staff than management when it comes to agility. About 60% of the ones who's worked here for a long time are still in a form of waterfall state of mind. (DL-B)

When another project manager were asked if teams could be given more autonomy and discard the overarching waterfall model they answered;

Yes if they really could take responsibility for the bigger picture. We are a bit behind that at the moment. The teams love to look sprint to sprint... If communication between teams would work I could let them be even freer and take responsibility for both the short-term and long-term. (PM-B)

The communication aspect of hybrid methodologies was a cornerstone of the responses of the respondents where the team members (DL-A, DL-B) in particular praised team rooms where developers co-create with testers. A team member summarizes their feelings on the subject as;

Partly developing an understanding for each other's work and difficulties but also developing an understanding for problem solving. There is an increase in consulting each other and how they want me to formulate issues for them to understand. When you don't communicate it becomes a 'them vs. us' mentality. There is less individualism and more of a team spirit, which allows you to take others opinions into consideration. (DL-A)

The use of team rooms and cross-functional teams in development is to increase communication but also increase freedom and work enjoyment;

A common discovery that many employees do in cross-functional teams is that they awaken. They become inspired, they think it is fun; they are all in the same room. It took a lot of persuasion for everyone to try it, but everyone who has loved it. Sure it might be noisy but all the positive things overshadow it. (DL-B)

This increase in personal responsibility and workflow freedom is a staple of agility and is deemed a more humane way of working;

Agile is definitely more humane. What appeals to me is the view of the individual. You give the teams mandates and let them organize it themselves. We have this need. The teams can do it. It is something completely different than when you get a thick bunch of demands. (SM-B)

The agile part of hybrid methodologies is centered on communication and self-organizing teams, following the definition of agility as stated earlier. This communication has in turn given the tester a higher status through agile work in comparison to traditional development according to a project manager (PM-C) a notion a developer level employee is in agreement with;

Through better communication you are more inclined to talk to one another and communicate across borders. It used to be a divide between developer and tester, you had no contact. Project management just issued demands. That is one of the things that have improved the most. You respect each other's professions and the knowledge you bring to the project. (DL-A)

When asked if the respondents would like to continue working in a hybrid methodology they had several different answers. Some stated they would if some criteria were met;

For bigger projects and for big time-commitments. For small product developments and improvements of existing products I think you can still work traditionally agile and yet add clarity in the way it is structured. (PM-B)

While three others simply hoped for increased agility (SM-A, PM-C, PM-D) and when asked if they would seize the chance to work within pure agility they all said yes. A developer level employee stated;

I think the waterfall influences will be phased out. Some think the old ways will return, but I don't think so. It works in limited areas, but not as a general solution. You become vulnerable with waterfall development, nothing unexpected can happen. The biggest issue is the economy, how you should get paid. (DL-B)

7. Analysis

The interviews gave significant findings regarding the motivation of the practice of hybrid methodologies and in order to get a clearer view of how they were enacted they will be placed within the ITMF in this section. To protect the privacy and anonymity of the participants within the study the analysis will be overarching conglomerate 1 and conglomerate B, even if individuals from different locations and organizations have been interviewed in one of the conglomerates.

7.1 The request stage

With no respondents explicitly mentioning the process within this business level buy-in for the need of the project it could indicate them not being operative in this stage. Although there is evidence that this stage could be causing issues for build stage. The notion of the sales department not fully understanding the inner workings of agility could indicate a waterfall based model within the request stage, where once completed it is not revisited. Another instance is respondent being presented with the requirements in the next stage, indicating that the client themselves analyzed their own need for change independently.

This stage is thus outside of the control of the project managers and someone outside of the agile development team performs the initial contact with the client. Within the ITMF neither waterfall nor Scrum are concerned with this stage, leaving it blank on a developmental view of the project. Since there are issues both with the sales department as

well as the economical reporting within the agile development this stage could be creating issues for the build stage if it is not aware of how development is conducted.

7.2 The define stage

The define stage is a more turbulent stage of the development, since differences of practice within it are present even within a single organization. There are influences of agility where developers are cooperating and co creating requirements; *"They (the developers) have been involved earlier and developed suggestions for solutions. But that was three years ago."* and there are instances where requirements are provided at a high abstraction level where they are defined parallel to the build stage. There are also evidence indicating that the requirements are important to the success of the project; *"If you haven't received a well constructed requirement specification from the client when you enter the project... I mean shit in, shit out. That's the truth."* This is an indication that the client themselves are providing the definition of the requirements, a habit that is hard to shake;

The big project compared with the messy teams is very waterfall based with handovers. Other teams are in the middle. In the refinement process developers are involved with product owners, interpreting requirements. The whole scale still exists within the organization.

A respondent clearly states that the feasibility study which is a staple of traditional waterfall development within the define stage is a remnant of the past, which might be true within their specific project but does seem to exist in different forms overlooking all the different cases. The requirements are most commonly defined on a high abstraction level in order to facilitate a dialog in the build stage;

We received a requirement specification in the beginning but the requirements were defined on a high abstraction level and we had the opportunity to work with questions that allowed for a dialog with the client.

With the agile parts of the methodologies being prototype based the design and planning within the define stage are overlooked in favor of a closer working relationship with the client; *"You can't plan for three months and execute that plan for three months, install at the client and wait for the disaster. That way of thinking is gone."*

This stage is thus a hybrid form of practice with influences from waterfall in the sense that the client could independently provide the requirements, but at a high abstraction level to facilitate dialog in the latter stages. It is also indicated that the requirements are set, but negotiable, a clear hybrid form of agility and waterfall. This stage is completed faster than traditional waterfall but the requirement handling is a hybrid within their current form. The design phase within the define stage is also bypassed in order to have a prototype delivery in the build stage, indicating further movement towards agility and for increased customer interaction.

This follows the model suggested by Hayata and Han (2011) which is initially waterfall based, as well as the critical success factors by Chow and Cao (2008), of having a correct delivery strategy. Having the requirements set but negotiable allows a higher degree of

control, according to a project manager, and could be an attempt to compensate for a lack of customer interaction.

7.3 The build stage

All the respondents contacted were part of mature teams working with Scrum and using sprint burndowns through a sprint backlog that closely followed the formalized Scrum method of development. They did not have a formalized design phase but instead chose to create a prototype to start cooperating with their client. According to Scrum the developers themselves plan their own sprints, and while this is the overall standard there is one respondent that refute the claim of working within Scrum the following way;

We say we use Scrum but I can tell you that we don't. We have picked parts out of it and we are getting closer and closer towards the method that is Scrum, but we aren't there yet.

This discrepancy of the agile concept is not confined to a single organization it instead noticeable in other areas. Another respondent states the following;

We have a lot more mature staff than management when it comes to agility. About 60% of the ones who's worked here for a long time are still in a form of waterfall state of mind.

Despite this dissension there is a complete agreement by all respondents that communication with the client is important, as well as communication within the project. Continuous testing as well as team rooms has increased communication, and in extent agility, giving developers shorter feedback loops. This has in turn allowed developers to shorten their reaction time to risks. Couple this reactivity with increased communication with the client and this has led to developers being more flexible and following the agile development way of revisiting finished parts of the project to increase customer satisfaction. This gives credence to Chow and Cao (2008) and their critical success factors, although there seems to be a higher focus on the other factors and thus diverging from the three main critical factors.

While there exists dissension within the build stage of how it is executed it seems that it is closer coupled with agility rather than traditional waterfall methodology. The way that agility is enacted in the cases is in line with the agile genome (Glaiel et al, 2013) and how agility is commonly enacted as feature driven incremental development containing cross-functional teams. The cases lack the refactoring and automated testing that made agility superior Glaiel et al (2013) experiment.

Despite this the testing is the factor that is the clearest representation of the hybrid form, combining both continuous testing with a clear testing phase at the end of the project. All project managers praise continuous testing that simplifies risk management and increase communication in the teams but some respondents still assert that a clear testing phase after the project is a necessity.

7.4 The deployment stage

After a clear agile phase within the build stage according to the majority of the respondents they turn into a testing phase. This testing phase can be iterative in the sense that many perform testing after each assigned deliverable deadline, as well as a complete testing phase after the project. Since the deployment of the project product or service is incremental in its nature, due to agility within the build stage, this stage is often performed iteratively as well as after the project. Meaning that since the deployment stage launches an artifact it follows development, but maintain a hybrid form. Agility claims testing during the project rather than after is superior, and as such the final deployment stage after the iterations is waterfall at its core. Since testing is both continuous as well as iterative with testing phases before deadlines and with a clear testing phase after a project this stage is a hybrid form of development.

The reason for this structure stems from the respondents unwillingness to release a product without a full system test, which the respondent with the clearest testing phase expressed confusion over while wishing for an increase in autonomous testing. It should be noted that autonomous testing is not part of Scrums methodology and as such could not be expected to be part of the projects. It should also be noted that since the overarching framework is an external effect the testing stage could be performed in order to appease the client. The continuous testing is a clearly an agile inspiration in order to both create cross-functional teams as well as increase customer interaction.

7.5 The run stage

The run stage is not connected to either waterfall or Scrum within the ITMF but acts as ownership transferal and closing down of project teams in combination of arranging the ongoing support of the product or service provided as a result of the project. Two respondents touched directly upon the support stage after the close of the project and one explained how they initially used Kanban boards but switched over to Scrum during this ongoing support stage to divide work after the closure of the project. The other respondent explained that while support is not currently directly a part of their build stage they will later be incorporated in order to prepare them for their work within the run stage. While there is not enough data to give a general view of the run stage across the conglomerates at least two organizations is using an agile method within this stage. Their reason for using it seems to be from both an increase in customer satisfaction as well as to distribute competence within the team. The latter was from Kanban since it forced participants to do different things from the Kanban board, thus avoiding specialization.

8. Discussion

Through my analysis of my collected data I feel I can answer my research question; *How is hybrid agile methodologies enacted in practice and why are they enacted in such a way?*

8.1 Agile in action

The result and analysis of the data showcase a combination of agile and waterfall work practices which are a result of a mixture of external factors influencing the development as well as traditional development ways of thinking not being fully retired from all participants.

Connecting these findings to West's (2011) study regarding the change of agility in practice in the form of mixing methods showcases that mixing agile practices is not as common as retaining certain waterfall aspects while managing the project. The motivation for using agility in development is clear, it increases communication with more humane working conditions as well as allowing risks to be seen at an earlier stage. The change of agility in practice into hybrid methodologies on the other hand is a way to compensate for external factors that influence the project from being completely agile. Thus the practice of hybrid methodologies and the change of agility in practice came primarily in relation to the overarching waterfall framework and not the mixing of agile methods.

The overarching waterfall framework is a solution to projects having issues with client interaction (Krasteva and Ilieva, 2008) and other external factors that affect development and consequently dictating how the methodology is enacted. Issues regarding economic reporting which does not follow the agile workflow in combination with the sales department not being educated in agility indicate that the hybrid form with a waterfall framework might not be a choice but rather a necessity. With agile methods being viewed as tools in a toolbox (Cannizzo et al, 2008; Glaiel et al, 2013) this study indicate that the tools available to development are heavily influenced by the surrounding organizational environment and thus partly outside of the management within development's hands. The overarching framework could with this view therefore be management's safeguard towards external factors on the project, allowing work to continue if for instance communication with the client is lacking.

While there exists dissent of how far the teams have traversed into agility all respondents praised agile practices. Despite this the managers seemed to agree that some form of requirement tracing is needed within the projects, indicating that this is easier within the overarching waterfall framework. This could be management's view of a correct delivery strategy (Chow and Cao, 2008) but responding to the external factors with a waterfall solution places the projects within the same common adoption efficiency suggested by Glaiel et al (2013) which further hinders the projects from reaching the potential that complete agile adoption brings.

The response of using an overarching waterfall framework could be an indicator of management maintaining a waterfall thought process and while there are traces of two cultures within the projects, it is unlikely this is due to management wishing to maintain a waterfall model. While it would be convenient to claim that the ambidextrous view proposed by Vinekar et al (2006) is applicable to the situation it is probable that the overarching culture is management's attempt to solve issues with external factors. To further provide credence to this there exists no evidence of intentionally dividing teams or management into waterfall and agile practices, on the contrary the use of cross-functional teams are encouraged in the projects, a clear departure from the ambidextrous view proposed by Vinekar et al (2006). This in combination with the respondents wishes for increased agility rather than waterfall practices further indicates that there are external factors rather than cultural factors that prevent them from increasing their methodological agile practices.

Using the ITMF showcased that the execution of these hybrid methodologies resembles Hayata and Han's (2011) hybrid model with waterfall at front into agile development followed by waterfall testing. A divergence from the model can be seen in the define stage in the form of several different practices which combine both agility with changing requirements but also practices of set requirements. While there are differences in the work practices within the cases it is clear that respondents maintain the overarching waterfall framework to adapt to external factors.

Thus the models suggested by Mushtaq and Qureshi (2012), Rasool et al (2013) and Castilla (2014) all lack an understanding of the external factors which influence agility within their studies and consequently missed an essential factor when it comes to the practice of methodologies. Even the controlled case study conducted by Nisa and Qureshi (2012) does not take into account the initial presentation of the project to the client through the sales department which sets the stage for development as well as the economic reporting which further affects the practice of the methodologies.

9. Conclusion

While there are differences of how the practice of the methodologies are enacted an overarching framework is found. This framework is in many ways deemed a necessity by several of the respondents, but with an almost unanimous wish for increased agility. The practice within the cases in this study is similar to the model presented by Hayata and Han (2011) where areas outside of the build stage have strong tendencies towards waterfall. As such the stages are waterfall/hybrid for requirements before moving onto agile development ending in a waterfall testing, while still performing continuous testing during the sprints in the build stage. External obstacles such as clients and surrounding organizational actors not being ready to cooperate to the extent which agility demands force the project into this hybrid methodology. Obstacles within the project such as upper management being unwilling to release control to developers affect the methodology, but external obstacles have through this study shown a greater effect on the practice of development methodologies. The portrayal from the sales department in the initial request stage coupled with traditional economic reporting during the project are obstacles which hinder projects from being completely agile and forces them into a hybrid mold in order to fit with the rest of the organization.

While the majority claim increase agility is desirable by the respondents there are also those who enjoy working within hybrid methodologies, claiming it gives them increased control over the project and thus allowing both flexibility and rigidity.

In order to fully understand why hybrid methodologies are used more research is required in the area, in particular on the factors outside of IT development. The relationship between IT development and other actors within the organization need to be explored since this study showcases that the obstacles between hybrid methodology and agile methodology are mostly external. While there exists evidence that a waterfall state of mind within development is a force of habit that affects development, which might be hard to break, the evidence pointing towards external factors is greater.

The use of the ITMF was suitable for the study and its use with hybrid methodologies was also the first of its kind as far as this researcher can tell. It showcased the discrepancies between the methodologies and allowed the analysis to be very concise by breaking the methodologies down into their different parts. With neither Scrum nor the waterfall model having processes in the request and deployment stage these stages were hard to analyze. Although they could be useful in explaining areas which might be deemed outside of development which affects it, for instance the sales department in the request stage.

To conclude the study many organizations may claim a complete adoption and use of agility in their methodology but are still affected by an overarching waterfall framework, a common occurrence according to this multiple case study. While this study acknowledges the existence of differences of agile adoption it still showcases a need for a greater understanding of the engagement between the external actors and software development in order to understand how agility is enacted in practice. It is possible that the historical divide between IT and the rest of the organization (Hoyer and Stanoevska-Slabeva, 2009) is still present within IT development practices.

10. Acknowledgements

I want to thank my respondents and their organizations for providing me the opportunity to study their methodologies. I also want to give my sincere thanks to my supervisor Rickard Harr for his knowledge and support throughout this thesis.

11. References

- Abrahamsson, P., Conboy, K. and Wang, X. (2009). 'Lots done, more to do': the current state of agile systems development research. *European Journal of Information Systems*, 18(4), pp.281-284.
- Agilemanifesto.org, (n.d.). Principles behind the Agile Manifesto. [online] Available at: <http://agilemanifesto.org/principles.html> [Accessed 9 Apr. 2015].
- Ahlemann, F., El Arbi, F., Kaiser, M. and Heck, A. (2013). A process framework for theoretically grounded prescriptive research in the project management field. *International Journal of Project Management*, 31(1), pp.43-56.
- Avifson, D. and Fitzgerald, G. (2006). Methodologies for Developing Information Systems: A Historical Perspective. The Past and Future of Information Systems: 1976–2006 and Beyond, pp.27-38.
- Avison, D. and Fitzgerald, G. (2006). Information systems development. 4th ed. Oxford [England]: *Blackwell Scientific Publications*.
- Björkholm, T. and Brattberg, H. (2010). *Prioritera, fokusera, leverera*. Stockholm: Crisp.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), pp.64-69.
- Boehm, B. and Turner, R. (2004). *Balancing agility and discipline*. Boston: Addison-Wesley.
- Boehm, B. and Turner, R. (2005). Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Softw.*, 22(5), pp.30-39.
- Bryman, A. and Nilsson, B. (2011). *Samhällsvetenskapliga metoder*. Malmö: Liber.
- Cannizzo, F., Marcionetti, G. and Moser, P. (2008). Evolution of the Tools and Practices of a Large Distributed Agile Team. *Agile 2008 Conference*.
- Castilla, D. (2014). A Hybrid Approach Using RUP and Scrum as a Software Development Strategy. Master's Thesis. College of Computing, Engineering & Construction.
- Christopher, M. and Towill, D. (2001). An integrated model for the design of agile supply chains. *International Journal of Physical Distribution & Logistics Management*, 31(4), pp.235-246.
- Chow, T. and Cao, D. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), pp.961-971.
- Cicmil, S., Williams, T., Thomas, J. and Hodgson, D. (2006). Rethinking Project Management: Researching the actuality of projects. *International Journal of Project Management*, 24(8), pp.675-686.
- Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3), pp.329-354.

- Dingsøy, T., Nerur, S., Balijepally, V. and Moe, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), pp.1213-1221.
- Dorairaj, S., Noble, J. and Malik, P. (2012). Understanding lack of trust in distributed agile teams: a grounded theory study. *16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)*.
- Dybå, T. and Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), pp.833-859.
- Emam, K. and Koru, A. (2008). A Replicated Survey of IT Software Project Failures. *IEEE Softw.*, 25(5), pp.84-90.
- Erdogmus, H., Morisio, M. and Torchiano, M. (2005). On the effectiveness of the test-first approach to programming. *IEEE Trans. Software Eng.*, 31(3), pp.226-237.
- Falessi, D., Oliveira, R., Taylor, K., Fontana, R., Power, K., Vallon, R., Giardino, C., Rejab, M. and Wang, X. (2014). Trends and emerging areas of agile research. *SIGSOFT Softw. Eng. Notes*, 39(5), pp.26-29.
- Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning. (2002). Vetenskapsrådet.
- Hass, K. (2007). The blending of traditional and agile project management. *PM World Today*, 9(5), pp.1-8.
- Hayata, T. and Han, J. (2011). A hybrid model for IT project with Scrum. *Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*.
- Hochmüller, E. and Mittermeir, R. (2008). Agile process myths. *Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral - APOS '08*.
- Hossain, E., Babar, M. and Paik, H. (2009). Using Scrum in Global Software Development: A Systematic Literature Review. 2009 *Fourth IEEE International Conference on Global Software Engineering*.
- Hoyer, V. and Stanoevska-Slabeva, K. (2009). The Changing Role of IT Departments in Enterprise Mashup Environments. *Service-Oriented Computing – ICSOC 2008 Workshops*, pp.148-154.
- Jalali, S. and Wohlin, C. (2010). Agile Practices in Global Software Engineering - A Systematic Map. 2010 *5th IEEE International Conference on Global Software Engineering*.
- Jiang, L. and Eberlein, A. (2008). Towards a framework for understanding the relationships between classical software engineering and agile methodologies. *Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral - APOS '08*, pp.9-14.
- Knoblauch, H. (2005). Focused Ethnography. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 6(3).

- Krasteva, I. and Ilieva, S. (2008). Adopting an agile methodology. *Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral - APOS '08*.
- Lozo, G. and Jovanović, S. (2012). A Flexible Hybrid Method for IT Project Management. *Journal of Emerging Trends in Computing and Information Sciences*, 3(7), pp.1027-1036.
- McHugh, O., Conboy, K. and Lang, M. (2012). Agile Practices: The Impact on Trust in Software Project Teams. *IEEE Softw.*, 29(3), pp.71-76.
- Miles, M., Huberman, A. and Saldaña, J. (2014). *Qualitative data analysis*. 3rd ed.
- Moe, N., Dingsøyr, T. and Dybå, T. (2009). Overcoming Barriers to Self-Management in Software Teams. *IEEE Softw.*, 26(6), pp.20-26.
- Mushtaq, Z. and Qureshi, M. (2012). Novel Hybrid Model: Integrating Scrum and XP. *International Journal of Information Technology and Computer Science*, 4(6), pp.39-44.
- Nerur, S. and Baliyepally, V. (2007). Theoretical reflections on agile development methodologies. *Commun. ACM*, 50(3), pp.79-83.
- Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Commun. ACM*, 48(5), pp.72-78.
- Nisa, S. and Qureshi, M. (2012). Empirical Estimation of Hybrid Model: A Controlled Case Study. *International Journal of Information Technology and Computer Science*, 4(8), pp.43-50.
- Patton, M. and Patton, M. (2002). *Qualitative research and evaluation methods*. Thousand Oaks, Calif.: Sage Publications.
- Pollard, J. and Geisler, S. (2014). Controlling a Permanent State of Change – IT Management Framework (ITMF). *JOEBM*, pp.48-52.
- Rasool, G., Aftab, S., Hussain, S. and Streitferdt, D. (2013). eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects. *JSEA*, 06(09), pp.446-457.
- Stolberg, S. (2009). Enabling Agile Testing through Continuous Integration. *2009 Agile Conference*.
- Vinekar, V., Slinkman, C. and Nerur, S. (2006). Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View. *Information Systems Management*, 23(3), pp.31-42.
- Ward, C. and Legorreta, L. (2009). Beyond Waterfall and Agile Methods: Towards a New Contingency Model for IT Project Management. *SSRN Journal*.
- West, D. (2011). Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. *Forrester Research*. [online] Available at: <http://www.storycology.com/uploads/1/1/4/9/11495720/water-scrum-fall.pdf> [Accessed 14 Mar. 2015].
- West, D. and Grant, T. (2010). Agile Development: Mainstream Adoption Has Changed Agility. *Forrester Research*. [online] Available at:

http://pmshow2012.programmedevelopment.com/public/uploads/files/forrester_agile_development_mainstream_adoption_has_changed_agility.pdf [Accessed 31 Mar. 2015].

Winter, M., Smith, C., Morris, P. and Cicmil, S. (2006). Directions for future research in project management: The main findings of a UK government-funded research network. *International Journal of Project Management*, 24(8), pp.638-649.

Yin, R. (1994). *Case study research*. 2nd ed. Thousand Oaks, Calif.: Sage Publications.