

EXPLORING ENHANCEMENTS TO THE AGILE APPROACH FOR MID-SIZED ENTERPRISES IN THE SERVICES SECTOR

L. Fourie^{1*#} & M. de Vries¹

ARTICLE INFO

Article details

Presented at the 28th annual conference of the Southern African Institute for Industrial Engineering (SAIIE), held from 25-27 October 2017 in Vanderbijlpark, South Africa

Available online 22 Nov 2017

Contact details

* Corresponding author
Lyndall.fourie@gmail.com

Author affiliations

¹ Department of Industrial and Systems Engineering, University of Pretoria, South Africa

The author was enrolled for an MEng (Industrial) degree in the Department of Industrial and Systems Engineering, University of Pretoria.

DOI

<http://dx.doi.org/10.7166/28-3-1837>

ABSTRACT

Companies need to thrive in a dynamic environment, responding to it in an agile way. Within a company's ICT domain, a key to achieving agility is to use agile software development. Experience, together with a literature review, has shown that the agile software development family of methods does not naturally fit into the context of medium to large enterprises. This article focuses on the challenges associated with agile software development. The main contribution of the article is to affirm that a class-of-problems exists - i.e., that many practitioners face challenges in implementing existing agile software development approaches successfully in medium to large enterprises. We also gather data via multiple interviews to show that the class-of-problems exists in a real-world enterprise that currently applies agile software development in medium-sized governmental bodies.

OPSOMMING

Ondernemings moet oorleef in 'n dinamiese omgewing deur pro-aktief op hulle omgewing te reageer. Ratse ('agile') sagteware ontwikkeling speel 'n sleutel rol binne die onderneming se inligting tegnologie afdeling om dit te bewerkstellig. Ondervinding, en 'n literatuurstudie, het aangedui dat ratse sagteware ontwikkeling nie natuurlik pas binne die raamwerk van groot en middelslag ondernemings se inligting tegnologie afdelings nie. Die artikel bespreek die uitdagings wat verband hou met ratse sagteware ontwikkeling, en bevestig dat 'n generiese probleem bestaan - naamlik, dat praktisyns probleme ondervind om ratse sagteware beginsel te implimenteer by groot en mediumsag ondernemings. Die studie maak gebruik van onderhoude om te bevestig dat die generiese probleem ook bestaan by werklike medium grote staatsinstellings wat tans ratse sagteware ontwikkeling toepas.

1 INTRODUCTION AND BACKGROUND:

This section provides background on how agile software development supports enterprise agility, and gives an overview of the concept of agile software development and the limitations and challenges associated with it. The section further provides the focus and structure of the article.

1.1 Enterprise agility

Enterprise agility is an important determinant of the success of an enterprise in a turbulent and dynamic environment [1]. Tsourveloudis and Valavanis [2] defined 'enterprise agility' as "the ability of an enterprise to operate profitably in a rapidly changing and continuously fragmented global market environment by producing high quality, high-performance, customer-configured goods and services".

A study conducted by Panda and Rath [3] has found that information technology (IT) capability acts as an enabler for enterprise agility. The study emphasised that significant investment in information technology will impede overall enterprise agility if the investment cannot be translated into a high

quality IT capability. Enterprises need continuously to nurture and develop superior enterprise-wide IT capability to ensure enterprise agility [4].

According to Dybå and Dingsøyr [5], the question of how software should be developed to deliver faster, better, and cheaper IT solutions has been discussed in software engineering circles for decades. Recently, the suggestions for improvement have come from experienced practitioners, who have labelled their methods ‘agile software development’ (ASD). Their methods have a big impact on how software is developed worldwide.

1.2 Agile software development

According to Shore and Warden [6], agile software development is more than a method: it is a philosophy. It is a way of thinking about software development, and is described by the four values and twelve principles that form part of the ‘agile manifesto’. According to Beck *et al.* [7], the agile software development manifesto emerged from a meeting held in 2001, attended by representatives from various software development disciplines. They identified a need for an alternative to document-driven, heavyweight software development processes. To survive and compete in the era of the internet, companies should focus on delivering what matters to the customer in a timely, tangible and ‘as promised’ way [7].

Turk, France and Rumpé [8] presented a paper on the limitations of the agile software development process, based on analysing published works on agile software development processes. Agile processes are difficult to assess in general terms, as they vary in values, practices, and application domains. Consequently, they decided to study the agile processes against the Agile Alliance principles and the assumptions that underpin a number of the more common agile software development methods, such as Extreme Programming and Scrum.

Based on their analyses, they have identified the following limitations: (1) limited support for distributed development environments; (2) limited support for subcontracting; (3) limited support for building re-usable artefacts; (4) limited support for development involving large teams; (5) limited support for developing safety-critical software; and (6) limited support for developing large, complex software applications.

Dikert, Paasivaara and Lassenius [9] did a systematic literature review on large-scale agile software development transformation challenges. They concluded that agile software development methods were originally designed for small and individual project teams, and that they therefore present challenges when introducing agile software development methods at scale. They identified nine main challenges: (1) it is difficult to manage dependencies between agile software development teams and the broader enterprise; (2) the goals of the agile software development teams and the broader enterprise are often mis-aligned; (3) it is difficult to maintain technical consistency between different agile software development teams; (4) using traditional and new agile methods side-by-side in an enterprise creates conflict; (5) silos that exist within a company make the implementation of agile methods difficult; (6) the majority of software agile development methods lack management of high-level requirements; (7) ambiguity about requirements influences project quality from the perspective of testing functionality; (8) there is a misunderstanding of the basic agile concepts that underpin the agile manifesto when adapting agile principles; and (9) it is difficult for some teams to design for the full picture when an incremental approach is followed.

The problems identified in the literature were supported by the researcher’s experience while working on agile software development projects in larger enterprises. These showed that the agile software development family of methods does not naturally fit into the context of medium to large enterprises. Companies can benefit from the advantages provided by the agile software development methods if they use an adapted agile approach that addresses the complexity of medium to larger enterprises successfully.

Both the literature review and a problem assessment conducted at a particular company identified a key area of concern: ineffective management of requirements within the enterprise. A study was initiated to answer the following research question:

What enhancements to the agile approach will ensure that relevant internal and external stakeholder requirements are addressed within agile timescales, and are useful when applied by the agile teams in a mid-size enterprise within the services sector?

1.3 Focus of the article

A prerequisite for answering the research question is to validate that there is an existing class-of-problems - i.e., that many practitioners face challenges in implementing existing agile software development approaches successfully in medium to large enterprises. This article focuses on validating the class-of-problems in general, and at a real-world enterprise that currently applies agile software development in medium-size governmental bodies.

1.4 Structure of the article

The research method used to validate the class-of-problems is explained in Section 2, followed by a discussion of the quantitative and qualitative results of the problem validation in Section 3. Section 4 presents the results of two analyses (inductive analyses of existing literature and interviews), and section 5 provides an indication of future work.

2 RESEARCH METHOD

For the purpose of this article, secondary and tertiary data was gathered from the literature to identify the deficiencies of the agile development approach. In addition, primary data was gathered via interviews to confirm that similar deficiencies are evident at a real-world enterprise that currently applies an agile software development method in medium-size governmental bodies. Since both sets of data were qualitative in nature, thematic analysis was used to extract the prominent deficiencies of the agile software development methods. We elaborate on the data-gathering and analyses techniques in the sections that follow.

2.1 Interview guidelines

Marchall and Rossman [10] argue that qualitative research relies extensively on in-depth interviewing. Brinkman and Kvale [11] define an interview as a conversation that has a structure and purpose, using careful questioning and listening to obtain thoroughly tested knowledge.

Brinkman *et al.* [11] describe the seven stages of an interview that were followed during our study. During the *thematizing* stage, the purpose of an investigation is formulated. The second stage focuses on the *design* of the interview, taking into account all seven stages, as well as the intended knowledge-creation and the moral implications of the study. In stage three the interview is *conducted* based on the interview guide, while reflecting on the knowledge sought and the interview as context. The interpersonal relation of the interview situation is also considered during this stage. *Transcribing*, where the interview material is prepared for analysis, is the focus of stage four, and the *analysis* of the material receives attention during stage 5. During stage 6, the *validity, reliability, and generalisability* of the interview findings are determined, and the final stage communicates the findings of the study and the methods applied.

The interviews allowed for sufficient probing to discover the various contributing themes that induce problems when applying an agile approach within a medium-sized enterprise context. Below is a sample of the interview questions:

1. Tell me about the problems you experience when working on agile software development projects in the company.
2. What are some of the challenges that you experience with requirements management on the projects that you work on?
3. What are some of the social challenges that you experience within your agile team?

In this article we used thematic analysis, as described in section 2.2, to analyse the interview transcriptions, presenting the analysis results in the right-most columns of Table 3.

2.2 Thematic analysis

Guest, MacQueen and Namey [12] define thematic analysis as a type of inductive analysis of qualitative data that can involve multiple analytic techniques. Thematic analysis describes both implicit and explicit ideas, or themes within data. Codes are developed to represent the identified themes and to link them back to the raw data for further analysis. The analysis may also include code frequency comparisons, code co-occurrence analyses, and a graphical display of the relationships between codes within the data set. If we compare thematic analysis with grounded

theory, the latter involves the process of reading the verbatim transcripts, identifying possible themes, comparing and contrasting themes to identify structure among them, and build theoretical models, constantly checking them against the data [12]. Thematic analysis is similar to grounded theory, but development of a theoretical model is not mandatory. However, interpretations should still be supported by actual data.

The applied thematic analysis approach followed for this study is based on the guidelines described by Guest *et al.* [12]. We highlight three phases.

Phase 1: During the *planning and preparation* phase, the analytical objectives are defined in line with the purpose of the study. It is a short description of what needs to be achieved through the lens of the data, and also includes data quality considerations, resources, and time availability [12]. In section 3 we discuss the main objective of the study, namely validating a problem with the usefulness of agile software development within medium-sized enterprises. We also defend our data-gathering strategy, using both secondary data (the literature) and primary data (collected via interviews).

Phase 2: The purpose of this phase is to *identify themes* in the text and to refine the themes into codes that form the basis of a codebook. This is an iterative process where the codebook is enriched as the data is collected and analysed. Code grouping may influence the way in which codes are applied during thematic analysis. The most common groupings are according to *structure* and *content*. *Structural coding* uses a hierarchical approach to organise codes into logical groupings based on questions related to a particular focus area of inquiry, and so are structured in line with the way that data is collected. Structural coding provides the link between the data collected and the resulting evidence. *Content coding* involves extracting meaning from text that exemplifies the analytical objective. Since this article focuses on problem identification, only structural coding was used to verify that there is an existing class-of-problems. Section 3 explains how we identified structural codes, based on an existing study (a systematic literature review) related to agile software development.

Phase 3: According to Guest *et al.* [12], *data reduction* techniques can take many forms in the context of thematic analysis. They may involve limiting the data set to relevant items only, to the preparation of qualitative code summaries and matrices, to advanced cluster analysis, or to graphing techniques. Although this article focuses on validating the existence of problems when agile is applied to medium-sized enterprises, the frequency of particular problem areas could direct future work on extending the agile development approach. Section 3 indicates how we quantified code frequencies to emphasise the prominence of certain kinds of problems when agile is applied at medium-sized enterprises.

3 PROBLEM VALIDATION RESULTS

In this section we validate that there is an existing class-of-problems, based on a thematic analysis of the literature (presented in section 3.1). Furthermore, we validate that an existing class-of-problems is found at a real-world company, based on a thematic analysis of semi-structured interview data (presented in section 3.2).

3.1 Validation that there is an existing class-of-problems

Thematic coding was used to extract meaning from the literature to validate that there is an existing class-of-problems - i.e., highlighting prominent deficiencies of the agile software development method when applied to medium to large enterprises.

3.1.1 Literature inputs

An extensive literature search was conducted that included several databases (Scopus, IEEE Xplore, Springerlink, Web of Science) and key words (“Agile software development”, “Agile software development” AND “challenges” OR “problems” OR “concerns” OR “issues”). The results were filtered according to document type (conference papers, articles, and book chapters) and year published (2015 to 2017). The results were then filtered on the content level, excluding sources that did not focus on agile software development challenges, sources that described specific agile software development challenges, and sources that did not focus on agile software development in medium- to large-scale enterprises.

A list of relevant systematic literature review sources is set out in Table 1. The systematic literature reviews include themes derived from primary studies (or secondary studies in the case of [13]). The codebook that we developed started with the identification of themes from the tertiary study of Hoda, Salleh and Grundy [13]. The scope of that study was not limited to agile software development in medium- to large-scale enterprises, but the themes it listed relate to fields of research, and so it was useful for creating high-level abstractions as structural themes of our code book. The systematic literature review [9] includes themes based on challenges identified in primary studies in the literature, and these were used to extend the structural themes. Data that was used to construct qualitative code summaries was only obtained from [9].

Table 1: List of systematic literature reviews

Systematic literature review title	Published date	Authors	Ref	Objective of the systematic literature review (SLR)
Systematic literature reviews in agile software development: A tertiary study	2017	R. Hoda; N. Salleh; J. Grundy	[13]	The aim of the paper is to provide an overview of the SLRs on agile system development research topics. Twenty-eight SLRs were included in the paper [13].
Challenges and success factors for large-scale agile transformations: A systematic literature review	2016	K. Dikert; M. Paasivaara; C. Lassenius	[9]	The aim of the study is to present a systematic literature review on how agile methods and lean software development have been adopted at scale, focusing on challenges and success factors. Forty-two cases were included in the study [9].

Sources of primary studies that were published before 2015 were excluded from the study, since relevant sources of good quality would have been covered in the recent systematic literature reviews. Table 2 lists the relevant primary studies that were published between 2015 and 2016 and that were not referenced in the systematic literature review of Dikert *et al.* [9]. The studies were also included in the thematic analysis.

Table 2: List of primary studies

Document title	Published date	Authors	Ref	Type of study
Challenges and success factors for large-scale agile transformations - A research proposal and a pilot study	2016	M. Paasivaara; C. Lassenius	[14]	Pilot survey
Applying agile methodologies in industry projects: Benefits and challenges	2015	K. Kaur; A. Jajoo; Manisha	[15]	Case study
Agile adoption case study, pains, challenges & benefits	2016	A. Anwar A.A. Kamel E. Ahmed	[16]	Case study
Agile in public sector: Case study of dairy farm management projects	2016	N. Wisitpongphan; T. Khampachua	[17]	Case study
Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project	2017	A. Azanha; A. R. T. T. Argoud; J. B. de Camargo; P.D Antonioli	[18]	Case study
Issues and challenges of cost management in agile software development projects	2016	Z. Bin Mansor; J. Yahaya; R. Razali; S. Yahya	[19]	Exploratory study

3.1.2 Inductive analysis results

Following an iterative process, we used structural coding to develop a codebook, which is summarised in Table 3. Content associated with the structural codes was identified but not coded, since future work will focus on identifying reasons for prominent themes. Although Table 3 presents calculated frequencies of themes for both the literature data and the interview data, we only discuss the results of the interview data in section 3.2.

Table 3: Codebook with themes and theme frequencies based on the literature review and on the interview data

Code #	Code name	Short description	Frequency in SLRs	Frequency in primary study	Frequency in interviews
1	Agile adoption (Change management)	Change management challenges associated with the adoption of ASD.	53 (31%)	13 (25.5%)	3 (4.8%)
2	Agile human and social aspects	Social challenges of individuals and teams.	10 (5.8%)	2 (3.9%)	4 (6.3%)
3a	Agile principles	Challenges associated with the level of insight of the agile core values and principles as specified in the Agile Manifesto.	4 (2.3%)	1 (2%)	1 (1.6%)
3b	Agile method	Challenges with the understanding of the agile method(s), impacting its use and customisation.	18 (10.5%)	3 (5.9%)	6 (9.5%)
3c	Agile practices	Challenges related to agile practices.	4 (2.3%)	5 (9.8%)	8 (12.7%)
4	Agile skills and expertise	Agile skills and expertise challenges.	9 (5.3%)	5 (9.8%)	5 (7.9%)
5	Requirements management	Requirements management challenges.	25 (14.6%)	7 (13.7%)	14 (22.2%)
6	Agile and usability	Integrating agile and user-centric design (UCD).	NA	NA	NA
7	Agile in a globally distributed project	Challenges associated with projects that cross geographical boundaries.	3 (1.8%)	1 (2%)	0 (0%)
8	Agile adoption (Enterprise agility)	Challenges to adopt ASD as a result of a lack of enterprise agility.	28 (16.4%)	4 (7.8%)	2 (3.2%)
9	Collaboration and communication	Challenges associated with collaboration and communication.	8 (4.7%)	9 (17.6%)	13 (20.6%)
10	Architecture and design	Challenges associated with architecture and solution design.	0 (0%)	1 (2%)	4 (6.3%)
11	Product quality	Challenges associated with product quality.	9 (5.3%)	0 (0%)	3 (4.8%)

The codebook elaborated on each theme in terms of a *structural code and code name*, *short description*, *long description*, and *where not to use*. As suggested by Guest *et al.* [12], the main researcher validated the codebook by coding one of the source documents using the codebook. The researcher waited for a week and then re-coded the same source to assess the level of consistency when compared with the initial coding. Since the inter-coder agreement exceeded 80 per cent, the researcher continued coding for the remaining literature sources.

Since we have already presented a summary of the themes in Table 3, we now provide the *long description* for each of the twelve themes. We include typical examples and qualifiers to guide appropriate coding.

- *Agile adoption (change management)*: The theme focuses on challenges related to the lack of commitment and resistance to change in the enterprise. It further includes challenges related to the adoption of roles, but not to training. An example of a valid fragment in support of this theme is: “People reverting to the old ways of working” [9]. *Where not to use*: Fragments referring to training should not be coded, since these fragments refer to a different code, namely 4. Fragments referring to the social aspects of individuals and teams should not be

coded, since these fragments refer to a different code, namely 2. Fragments referring to the structure and complexity of the enterprise should not be coded, since these fragments refer to a different code, namely 8.

- *Agile human and social aspects*: The theme focuses on challenges related to motivational factors, social behaviour, and the configuration and management of the team from a human social perspective. The theme also includes aspects of organisational culture, but excludes communication and collaboration challenges. An example of a valid fragment in support of this theme is: “Self-organisation is messy and takes time to achieve” [19]. *Where not to use*: Fragments referring to agile adoption changes should not be coded, since these fragments refer to a different code, namely 1. Fragments referring to the combination of agile and traditional methods (e.g. waterfall) should not be coded, since these fragments refer to a different code, namely 3b. Fragments referring to collaboration and communication should not be coded, since these fragments refer to a different code, namely 9.
- *Agile principles*: The theme focuses on challenges related to the level of insight or understanding that users have of the agile core values and principles as specified in the Agile Manifesto. An example of a valid fragment in support of this theme is: “Misconceptions of what agile software development is” [9]. *Where not to use*: Fragments referring to lack of understanding of agile method should not be coded, since these fragments refer to a different code, namely 3b.
- *Agile method*: The theme focuses on interpretations of agile methods that can lead to poorly customised methods. The theme does not include situations where agile and traditional methods need to co-exist due to complex enterprise structures, but include situations where agile and traditional methods co-exist due to the way the method was customised. An example of a valid fragment in support of this theme is: “There simply was not a manual or document where we could find easy answers on how to do things” [9]. *Where not to use*: Fragments referring to people reverting to old ways of working should not be coded, since these fragments refer to a different code, namely 1. Fragments referring to situations where agile and traditional methods need to co-exist due to the complexity of the enterprise structure should not be coded, since these fragments refer to a different code, namely 8. Fragments referring to lack of agile knowledge due to lack of training or coaching should not be coded, since these fragments refer to a different code, namely 4.
- *Agile practices*: The theme focuses on challenges related to agile practices, such as effort estimation, metrics, budgeting, and progress monitoring and feedback. An example of a valid fragment in support of this theme is: “Without estimates, sprint planning had no significant base” [15]. *Where not to use*: Fragments referring to requirements management should not be coded, since these fragments refer to a different code, namely 5.
- *Agile skills and expertise*: The theme focuses on challenges related to the agile skill set, and covers training and coaching. An example of a valid fragment in support of this theme is: “Lack of agile knowledge was the major contributing reason for organisational resistance” [16]. *Where not to use*: Fragments referring to agile method should not be coded, since these fragments refer to a different code, namely 3b.
- *Requirement management*: The theme focuses on challenges related to requirements management. Examples are lack of high level requirements management, requirements traceability, and issues with translating requirements into user stories. The theme further includes challenges related to team autonomy where teams have different goals. An example of a valid fragment in support of this theme is: “High level requirements management largely missing in agile” [9]. *Where not to use*: Fragments referring to agile methods should not be coded, since these fragments refer to a different code, namely 3b.
- *Agile and usability*: This theme is not a challenge identified in the literature, but a research theme identified in Hoda *et al.* [13]. It was included in the codebook as a structural code, and relates to integration between agile and user experience (UX) and the use of artefacts to communicate the UX design. The theme is included in the codebook, as it will be used for future analysis purposes.
- *Agile in a globally distributed project*: This theme relates to challenges caused by teams spread across different geographical locations. An example of a valid fragment in support of this theme is: “Differences in time zones can further toil the situation” [15]. *Where not to use*: Fragments referring to collaboration and communication should not be coded, since these fragments refer to a different code, namely 9.
- *Agile adoption (enterprise agility)*: This theme relates to challenges in adopting ASD due to lack of enterprise agility, such as the complexity of the enterprise structures, workforce, infrastructure, and processes. An example of a valid fragment in support of this theme is:

“While introducing agile had created flexibility at the team level, the surrounding organisation was not responsive enough” [9]. *Where not to use:* Fragments referring to agile adoption due to change management should not be coded, since these fragments refer to a different code, namely 1.

- *Collaboration and communication:* This theme relates to challenges associated with collaboration and communication between resources within an agile team, between agile teams, between agile teams and the enterprise, and between agile teams and the customer. This theme includes challenges associated with documentation. An example of a valid fragment in support of this theme is: “Customers sometimes try to avoid getting involved directly” [19]. *Where not to use:* Fragments referring to challenges related to the social aspects of individuals and teams should not be coded, since these fragments refer to a different code, namely 2.
- *Architecture and design:* This theme focuses on challenges associated with architecture and solution design (designing for the big picture). An example of a valid fragment in support of this theme is: “Struggling with maintaining the big picture in the design when working in iterations” [9]. *Where not to use:* Fragments referring to product quality should not be coded, since these fragments refer to a different code, namely 11.
- *Product quality:* This theme focuses on challenges associated with product quality. Examples are technical debt and quality control/assurance challenges. An example of a valid fragment in support of this theme is: “Inconsistency in design standards” [9]. *Where not to use:* Fragments referring to requirements management should not be coded, since these fragments refer to a different code, namely 5.

The results from the literature thematic analysis are illustrated in Figure 1. The blue bars represent the results from the systematic literature review from secondary studies, and the grey bars represent the results of the thematic analysis from primary studies. The frequencies of theme occurrences are shown as percentages (y axis). The graph excludes the theme relating to *Agile and usability*.

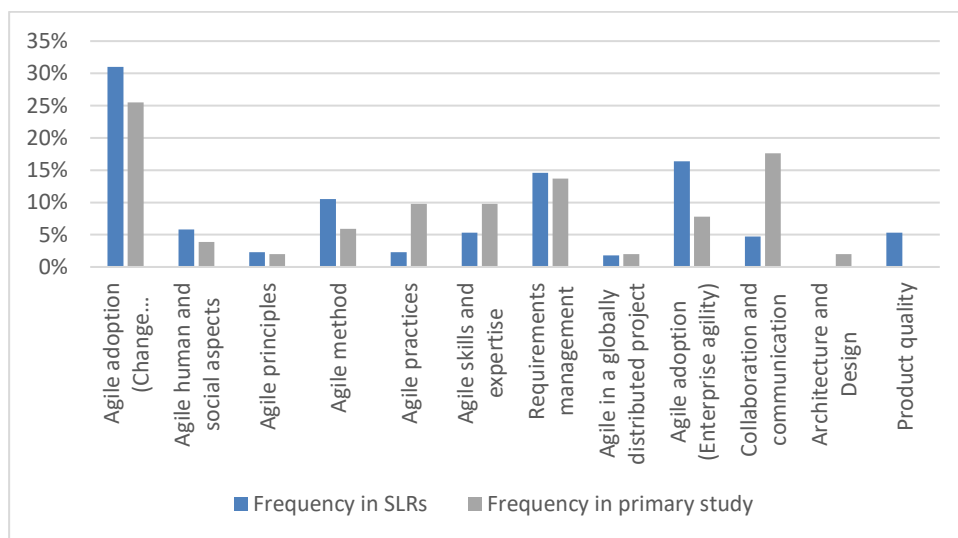


Figure 1: Graph illustrating the literature thematic analysis results

Agile adoption challenges due to change management is the theme that occurs the most in the literature thematic analysis for both the primary and secondary studies, followed by the themes that focus on *requirements management challenges* and *agile adoption due to enterprise agility challenges*. *Collaboration and communication* is a prominent theme in the primary studies.

Although the results indicate prominent challenges for medium- and large-sized enterprises, not all literature sources explicitly stated contexts such as industries, sectors, and developing vs developed enterprises. The literature study provided a broader context for discovering challenges when agile is applied within medium- or large-sized enterprises. Since other contextual factors may influence the success of an agile approach application, the next section presents the analysis results obtained from a real-world enterprise that delivers software solutions to customers across Africa, where agile is applied to medium-sized governmental bodies. The enterprise started its agile journey in 2014, and is fairly developed in the use and implementation of agile methods.

3.2 Validation that there is an existing class-of-problems in a real-world instance

Four structured interviews were conducted with two scrum masters, a scrum master who also fulfilled the role of a solution architect, and a business analyst manager. The results were used to confirm that similar deficiencies, identified as themes in section 3.1, also feature at a company that currently applies agile software development in medium-sized governmental bodies. The interview question, “*Tell me about the problems you experience when working on agile software development projects*”, was asked to gather the interviewees’ thoughts and concerns about agile software development at the company for which they worked. The employees were given the opportunity to frame and structure their responses freely, after which the interviewer used probing to extract more information about the challenges faced by the interviewee. The probes were structured according to the themes that emerged from the literature, as discussed in section 3.1. and formed the basis for structural codes. The structural code descriptions were enriched with identified content, and content coding will be applied in the next phase of problem analysis.

Figure 2 illustrates the thematic analyses results that are based on the interview data. *Requirements management challenges* and *collaboration and documentation challenges* are prominent themes, followed by *challenges associated with agile practices*. The results align with the thematic analyses of the literature, with the exception of *agile adoption due to change management or enterprise agility challenges*, which were not identified as being the leading challenges. There might be various reasons for this omission, such as a lack of commitment on the part of top management.

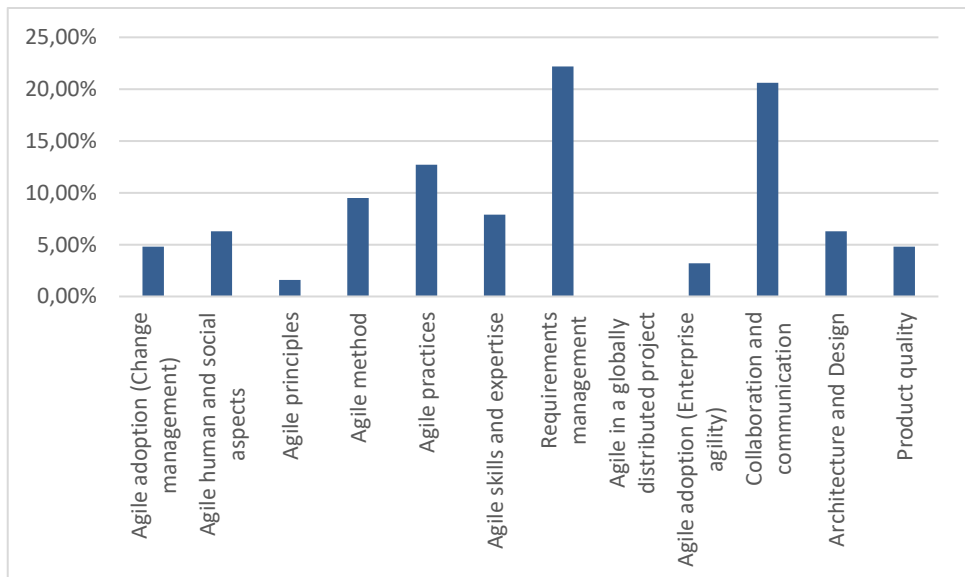


Figure 2: Graph illustrating the interview thematic analysis results

4 DISCUSSION

The thematic literature review has validated that there is an existing class-of-problems - that is, when agile development is used at medium to large enterprises, several challenges feature. Prominent challenges include: *agile adoption due to change management*, or *enterprise agility challenges*, *requirements management and collaboration challenges*, and *challenges associated with agile practices*.

The thematic interview analysis has validated that the existing class-of-problems identified in the literature review also exists as an instance in a real-world enterprise that currently applies agile software development principles. *Requirements management challenges* and *collaboration challenges* were identified as the dominant themes.

Although theoretical works specifically focus on medium or large enterprises, many other contextual factors, such as industry-type and level of enterprise maturity, can produce unique challenges in a more specific context. We acknowledge that, even though our analysis results from a real-world

mid-sized enterprise share similar challenges to those identified in theory, further data-gathering and analysis would be required to understand the impact of various contextual factors on the success of an agile implementation.

5 CONCLUSION AND FUTURE WORK

This article has focused on validating the existence of a general class-of-problems: that the agile software development approach poses several challenges when applied within the context of medium to large enterprises. An inductive analysis of selected literature sources confirmed the existence of the stated class-of-problems, and also highlighted prominent challenges. Furthermore, an inductive analysis of interview data, gathered from a real-world enterprise that currently applies agile software development in medium-size governmental bodies, confirmed some of the challenges.

Although the literature indicates that *agile adoption (change management)* is a prominent challenge, the interview data did not highlight this particular challenge. It would be interesting to inquire why this challenge does not feature in the real-world enterprise. *Requirements management* featured as a prominent challenge for both data sets (literature data and interview data), and could be studied further to identify the reasons for challenges in this area.

The current study has focused only on some key challenges when implementing agile in medium- or large-sized enterprises, not on discovering the underlying causes of these challenges. Future work will focus on extending the analysis to a larger sample of employees in the enterprise to validate the results discussed in this article. The technique of thematic content analysis will be applied to extract meaning from the text and to identify underlying themes that will provide more insight into the main challenges identified. During further analysis, a root-cause-analysis might highlight other causes and contextual factors that will direct the further development and refinement of the agile approach. The intent would be to enhance the agile approach, such that relevant internal and external stakeholder requirements are addressed within agile timescales, and are found to be useful when applied by the agile teams in a mid-sized enterprise within the services sector.

REFERENCES

- [1] Overby, E., Bharadwaj, A. & Sambamurthy, V. 2006. Enterprise agility and the enabling role of information technology, *European Journal of Information Systems*, 15(2), pp 120-131.
- [2] Tsourveloudis, N.C. & Valavanis, K.P. 2002. On the measurement of enterprise agility, *Journal of Intelligent & Robotic Systems*, 33(3), pp 329-342.
- [3] Panda, S., Panda, S., Rath, S.K. & Rath, S.K. 2016. Investigating the structural linkage between it capability and organizational agility: A study on indian financial enterprises, *Journal of Enterprise Information Management*, 29(5), pp 751-773.
- [4] Lu, Y. & Ramamurthy, K. 2011. Understanding the link between information technology capability and organizational agility: An empirical examination, *Mis Quarterly*, pp 931-954.
- [5] Dybå, T. & Dingsøy, T. 2008. Empirical studies of agile software development: A systematic review, *Information and software technology*, 50(9), pp 833-859.
- [6] Shore, J. 2007. *The art of agile development: Pragmatic guide to agile software development*, 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc.
- [7] Beck, K., Beedle, M., van Bannekum, A., Cockburn, C., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. 2001. *Manifesto for agile software development*, [accessed, available from: <http://agilemanifesto.org/>].
- [8] Turk, D., France, R. & Rumpe, B. 2014. Limitations of agile software processes, *arXiv preprint arXiv:1409.6600*.
- [9] Dikert, K., Paasivaara, M. & Lassenius, C. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review, *Journal of Systems and Software*, 119, pp 87-108.
- [10] MARSHALL, C. & ROSSMAN, G. 2011. *Design qualitative research. 5 th editon*, United States of America: Sage Publications.
- [11] Brinkman, S. & Kvale, S. 2015. Interviews: Learning the craft of qualitative research interviewing, *Aalborg*. Accessed January, 24, pp 2017.
- [12] Guest, G., MacQueen, K.M. & Namey, E.E. 2011. *Applied thematic analysis*: Sage.
- [13] Hoda, R., Salleh, N., Grundy, J. & Tee, H.M. 2017. Systematic literature reviews in agile software development: A tertiary study, *Information and Software Technology*, 85, pp 60-70.
- [14] Paasivaara, M. & Lassenius, C. 2016. Challenges and success factors for large-scale agile transformations: A research proposal and a pilot study, in *Proceedings of the Scientific Workshop Proceedings of XP2016 ACM*, pp 9.

- [15] **Kaur, K. & Jajoo, A.** 2015. Applying agile methodologies in industry projects: Benefits and challenges, in *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on IEEE*, pp 832-836.
- [16] **Anwar, A., Kamel, A.A. & Ahmed, E.** 2016. Agile adoption case study, pains, challenges & benefits, in *Proceedings of the 2nd Africa and Middle East Conference on Software Engineering ACM*, pp 60-65.
- [17] **Wisitpongphan, N. & Khampachua, T.** 2016. Agile in public sector: Case study of dairy farm management projects, in *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on IEEE*, pp 1-5.
- [18] **Azanha, A., et al.** 2017. Agile project management with scrum: A case study of a brazilian pharmaceutical company it project, *International Journal of Managing Projects in Business*, 10(1), pp 121-142.
- [19] **Mansor, Z., et al.** 2016. Issues and challenges of cost management in agile software development projects, *Advanced Science Letters*, 22(8), pp 1981-1984.