ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(30), DOI: 10.17485/ijst/2015/v8i30/87111, November 2015* ISSN (Online) : 0974-5645

Factors Limiting the Implementations of Agile Practices in the Software Industry: A Pilot Systematic Review

Salmiza Saul Hamid*, Mohd Hairul Nizam Mohammad Nasir, Mohd Khalit Othman and Rodina Ahmadi

Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia; salmiza.saulhamid@gmail.com

Abstract

This paper presents a pilot Systematic Literature Review (SLR) study which examined the limitations of Agile methods in the software industry based on primary research. The study investigated IEEE conferences and journal papers published between 2007 and 2012. 29 papers were identified as the most relevant. While the SLR findings brought to light limitations both in the Agile methods and in their implementation, the latter issues were the ones most frequently addressed. The results reveal that the three most repeatedly cited factors are high dependency on people/personnel, organisational dependency, and high impact on organisational structure and culture. While these three factors are mutually related, the people factor – especially the need for strong involvement and support from upper - level management – can be regarded as a primary necessity in Agile implementation. In spite of the Agile principles laying emphasis on people-critical functions, and in spite of the excellent rules in the gist of the principles, problems still arose when it came to the implementation stage. This finding indicates the need for future work on proper guidelines for management, given that existing guidelines for Agile adoptions and implementations are general and less focused on upper level managers.

Keywords: Agile Method, Limitation, Management, People Factor, Systematic Review

1. Introduction

Agile methods are now entering the mainstream. They are being adopted extensively outside their initial intended scope of small and collocated project teams. They are being implemented in numerous project sizes (small, medium and large), both in distributed (locally and globally) and non-distributed project environments, as well as in diverse project domains such as engineering, manufacturing, banking and medicine.

As a result, new versions of Agile method have been developed, and Agile methods have been enhanced or are being integrated with other models to support the increasing demands of different project environments. The initial intention was to integrate Agile Method (s) (AM) with an existing framework originating from outside the software industry to enhance the adoption coverage of AM. To explore the limitations of Agile practices further before deciding on their suitability for integration, a pilot Systematic Literature Review (SLR) study was conducted.

While the study aimed to identify the limitations of AM in the software industry generally, the most prominent limitations identified lay in the adoption and implementation of AM. The Agile Manifesto, which has been described as too informal[5] and embracing abstract principles[1], raised a lot of issues in the implementation and adaptation of Agile practices. Regardless of abundant suggestions and recommendations based on existing experiences, the flexibility and generality of AM left them open to various interpretations, and therefore invited difficulties in how to perform AM effectively in practice, notably for early adopters.

In this paper, we firstly highlight the background of this pilot SLR study encompassing; a brief introduction

* *Author for correspondence*

to Agile methodology, previous related studies, and the motivation behind the study. The review method or steps taken in performing the SLR is then elucidated. The results, consisting of an overview of the study and the compiled limitations, are presented, and last but not least, we conclude our SLR study with recommendations for future work.

2. Background

2.1 Agile Methodology

Agile methodology is an iterative and evolutionary approach for software project development. Initiated by seventeen experts or 'organizational anarchists' in February 2001, it operates under four core values and twelve principles, which are incorporated in the Agile Manifesto[7]. This widely known people-centric process model is inspired by sentiments such as 'working with people who shared compatible goals and values based on mutual trust and respect, promoting collaborative, people-focused organizational models and building the types of professional communities in which we would want to work'[7].

There are different types of AM – for example, the Dynamic Systems Development Method (DSDM), Extreme Programming (XP), SCRUM, Crystal Clear, Feature-Driven Development and Lean Software Development. Each AM has its own practices, which concentrate on different things. Thus, while XP concentrates on the project-level activities of software deployment, SCRUM 'concentrates on the management aspects of software development'[8].

2.2 Related Studies

The first systematic review study on AM, done by Dingsøyr and Dyba[6], intended to tackle the issue of anecdotal evidence in the Agile adoption success story. The SLR specifically reviewed studies of Agile software development from the beginning up to 2005, with search results from a total of 1996 papers. They discussed the benefits and limitations of AM as found in 36 completed empirical studies, as well as the strength of the evidence. Their inclusive analysis provides guidance and comparison, based on situational applicability, for an industrial readership. To conclude, their study mainly advocated further similar studies in the same area, but with better quality.

This next systematic review of Agile methodology was specific to the Global Software Engineering (GSE) environment[9]. The SLR included papers from five electronic databases from the years 1999 to 2009, and 77 of the most relevant papers were analysed. The study found that in most cases Agile practices were customised to fit the project environments and requirements. Thus, the study pointed to a requirement for future work to incorporate existing experiences in order to aid Agile adopters in distributed project settings.

Subsequently, Causevic et al[4] conducted a systematic review of factors limiting the industrial adoption of one of the Agile practices, namely, Test-Driven Development (TDD). This SLR study, examining both focused and non-specific TDD empirical studies, covered industrial and academic papers from 2005 to 2009. The study was based on earlier research findings that the practice of TDD was 'not followed to the extent preferred by industry'[4]. Based on the result, the paper addressed the need for guidelines 'to overcome these limiting factors for successful industrial adoption of TDD'[4]. While papers from seven electronic databases were investigated, the SLR focused on only one basic Agile software development practice.
The only paper with a similar focus to that of the present SLR study was by Livermore[12]. Livermore's study was, however, a worldwide online survey of general focus involving various industrial areas. With a low response rate of 5.76% or 112 survey responses, factors that were found to significantly impact Agile implementation included training, management involvement, access to external resources, and corporate size. Though the scope of its findings was smaller than that of the present SLR study, management was identified as having the main influence, ahead of the other factors.

Lastly, Neves et al[14] have presented a thorough analysis and empirical evaluation of the benefits and limitations of Agile practices experienced by Agile teams in relation to knowledge formation and transition. In the last phase of evaluation, a SWOT (Strength Weakness Opportunity Threat) analysis was utilised to evaluate Agile contributions to the productivity of the software development teams. The findings of the Neves paper show a number of limitation items similar to those identified in our own SLR. They result from weaknesses of and threats to Agile processes in knowledge management.

2.3 Motivation and Objectives

AM are at present getting wider attention and are being more widely used. However, early investigation (at the

end of 2011) found that their usage was limited. Despite their ambitious purpose, they were more suitable for experienced and skilled software engineers, and less appropriate for large and complex projects. AM are regarded as an ambitious practice because they allow for changes within a rapid software development environment, even when the anticipated outcome is yet to be stably attained. At the same time, no paper was found that inclusively discussed the drawbacks of AM.

As a result, a pilot SLR was conducted to dig out the limitations of Agile practices and prioritise them based on frequency. The SLR review process was very helpful, as the reviewer could clearly understand the situations and issues surrounded the practice of AM. This study also allowed the reviewer to see different angles in the research area and helped her to decide on the direction to focus on.

3. Review Method

3.1 Informal Review

First and foremost, an informal review was conducted to gain some understanding of the issues surrounding AM. In the process, the author reviewed articles, journals and conference papers by searching with Google and in several well-known electronic databases, such as the ACM portal, IEEE Xplore, ScienceDirect and SpringerLink, to uncover the limitations or constraints of AM. The information obtained from this preliminary search and review was used as a basis for planning the systematic review.

3.2 Systematic Review

To form an appropriate systematic review, a separate search on papers discussing systematic review methods was performed. Two combinations of search strings were used: "systematic review AND software" and "systematic review AND Agile". In the process, several papers were retrieved based on title, where the final selection were made based on the abstract and the date of publication. Two systematic review papers with different AM research focuses were chosen and are referred to here: Empirical studies in Agile software development[6] and Agile practices in global software engineering[9]. Their review techniques were carefully assessed and combined in building this systematic review plan. Their SLR methods, which followed several SLR guidelines (i.e. those in)[10], provided very helpful and clear-cut guidance for the reviewer to work on her own SLR.

3.2.1 Protocol Development

Basically, the design of this SLR review protocol was strongly influenced by the review methods presented by Dingsøyr and Dybå[6] and Jalali and Wohlin[9]. The combination of their review techniques resulted in the following construct.

3.2.2 Research Question

In a systematic literature review, research questions which are required to be built at the outset defines the scope of the review study for the paper. In this review study, the following research question was formulated with the objective of finding and gathering the limitations of AM:

What limitations of Agile practices have been reported/addressed to date?

3.2.3 Inclusion and Exclusion Criteria

For the searching and filtering process, the following inclusion and exclusion criteria were applied:

Inclusion criteria:

- Limitations of Agile practices.
- Conferences and journal papers (2007-2012).
- Findings from software industry.
- Research type: case study, empirical study, experience-based report, survey.
- Project size: all sizes (small, medium and large).
- Project type: distributed and non-distributed.

Exclusion criteria:

- Non-English literature.
- Findings from academic sources.

3.2.4 Search Strategy

A lot of previous research reports pointed out a problem with the validity of AM studies, noting most of the reported results were anecdotal. Based on the situation, the research community sought more concrete studies on AM, and more and more AM studies focusing on tangible results have subsequently been published. Therefore this SLR study started with searching AM-related papers that were based on a strong foundation, such as empirical or case studies, as well as surveys and industrial experience reports with concrete evidence or strong justifications. In the process, a reviewer examined the research method used in each paper to establish its validity.

Since this review study is about finding the limitations of Agile practices, the primary keyword used in the search

process was 'agile', with 'limitation' being the secondary keyword. For the first keyword, there were different types of AM, such as Extreme Programming (XP), SCRUM, Crystal Clear, Dynamic Systems Development Method (DSDM), Feature-Driven Development and Lean Software Development. A separate search was conducted to identify the limitations of each of these AM types. Since the second keyword "limitation" was a common noun, words of similar meaning were also listed. These words were spotted from papers in the earlier, informal review process. The formulation of several search strings from the keywords ensured that only related papers were extracted. The searching activity was conducted using IEEE Advanced Search, the sole electronic database used in this pilot SLR. IEEE has been one of the general sponsors for Agile Conference annual events since 2003 and is the main publisher of papers from the conferences.

Altogether, seven search strings were formed and used in the search process. This was to ensure that all the related papers, as determined by the research question, could be extracted. Basically, there were two types of search string, as indicated below:

First search string: to extract all papers related to 'Agile practices in software project' that had the 'Agile' keyword in their abstracts.

((((((((((“Abstract”: drawback) OR “Abstract”: challenge) OR “Abstract”:limitation) OR “Abstract”:issue) OR “Abstract”:difficulties) OR “Abstract”:problem) OR “Abstract”:constraint) OR “Abstract”:shortcoming) AND “Abstract”:Agile) AND “Abstract”:software)

Second to seventh search strings: to extract papers that discussed specific Agile practices such as SCRUM which were not included in the first search, where these papers did not have the 'Agile' keyword in their abstracts.

((((((((((“Abstract”: drawback) OR “Abstract”: challenge) OR “Abstract”:limitation) OR “Abstract”:issue) OR “Abstract”:difficulties) OR “Abstract”:problem) OR “Abstract”:constraint) OR “Abstract”:shortcoming) AND “Abstract”:scrum [/extreme programming/crystal clear/ dsdm/feature-driven development/lean AND software AND development]) AND “Abstract”:software) NOT “Abstract”:Agile).

3.2.5 Data Extraction

The reviewer went through four stages of searching and filtering to identify relevant papers (Figure 1). In the first stage, a total of 306 search results were returned after seven individual searches in IEEE Advanced Search. The next stages were a three-phase filtering process. While the first filter was mostly quite straightforward (scanning the titles or abstracts), there were cases where the reviewer needed to go beyond the abstract to understand further and decide. During the last two phases, the reviewer carefully went through each paper, in most cases needing to go through the whole paper or study the full text to identify its relevancy. In all three phases, the intended data as well as potential points were extracted and stored in an Excel file.

In the final stage, 29 papers were found to be the most relevant to the SLR topic. The subsequent process of data extraction was not straightforward, as the process was delayed by the way in which the studies/experiences were reported. The limitation elements were mostly pointed out only generally or non-transparently (not on the surface). Thus, the intended points could not be easily extracted and required critical reading and understanding. Furthermore, the points to be extracted were not fixed or unique, meaning that a keyword search was not possible. In short, the data on AM limitations were extracted based on the reviewer's analysis and conclusions after she had read the whole paper normally more than once in order to understand the cases/experiences reported.

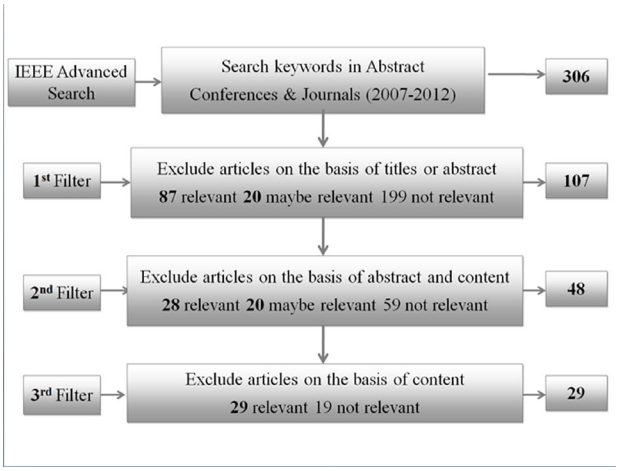

Figure 1. Stages in the study selection process.

4. Results and Discussion

4.1 Overview of Studies

Twenty-nine studies that addressed limitations in the AM implementations were identified. Table 1 illustrates the number of the related papers found in IEEE year by year. The constraints of AM were consistently addressed from 2007 to 2012 (the 8th of August 2012), even though the papers were simply extracted from an electronic database.

Fourteen papers were case studies of either single or multiple case studies, eleven were experience reports, three were empirical studies, and the remaining paper was a survey. 38%, or eleven papers, were confirmed as having large projects or large-scale Agile adoptions (Figure 2).

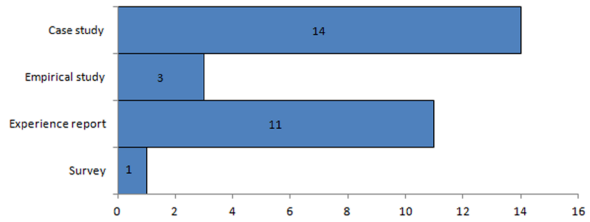

Figure 2. Related papers by research method.

Out of the 29 relevant papers (Table 2.), the overwhelming majority (90%) were published in conferences, while only three (10%) were published in journals. In this review study, SCRUM was found to be the most frequently adopted Agile technique (38%), followed by Agile practices in general (28%). The rest were Extreme Programming (XP) (2 studies), Lean (1 study), Lean and Agile (1 study), Evo (1 study) and a mixture of several AM (1 study). While SCRUM alone was adopted in most cases, there were also hybrid implementations of SCRUM and XP (1 study), and of SCRUM and User Stories, as well as SCRUM and Lean (1 study). A new approach in practising SCRUM, namely Enterprise SCRUM, was also developed and exercised (1 study). Our result aligns with several earlier reported studies indicating that at present SCRUM is the most favoured approach among other AM[S2,2,3,16].

Table 2. Type of Agile method used in the related papers

Agile method	Number	Percent
SCRUM	11	38%
Agile practices (general)	8	28%
XP	2	6.8%
SCRUM and XP	1	3.4%
SCRUM and User Stories	1	3.4%
Enterprise SCRUM and Kanban	1	3.4%
SCRUM and Lean	1	3.4%
Lean	1	3.4%
Lean and Agile (general)	1	3.4%
Evo	1	3.4%
Mixed	1	3.4%
Total	**29**	**100%**

Table 3. Related papers by project type

Project Type	Number	Percent
Distributed	6	21%
Distributed (GSD)	4	14%
Non-distributed	16	55%
Mixed	3	10%
Total	**29**	**100%**

More than half (55%) of the reviewed papers implemented Agile in collocated or non-distributed project settings, 21% applied Agile in distributed project environments, 14% discussed globally distributed Agile projects, and the remaining three studies (or 10%) reported Agile implementation in both environments (Table 3).

Five Agile transition cases were reported. Of these, four of involved large projects and/or large-scale Agile adoptions, and four of the five cases adopted SCRUM as well[S11,S16,S22,S25]. In addition, an experience report discussed how the SCRUM method had been extended to the executive level for the effective handling of large-scale SCRUM implementation[S10].

4.2 Limitation Factors

The limitations in Agile implementation were compiled into several categories, as shown in Table 4. High dependency on people/personnel, organisational dependency, as well as high impact on organisational structure and culture were found to be the three most important factors limiting Agile implementations in the software industry. In this sector, we highlight the issues identified from the related papers on the three factors.

4.2.1 High Dependency on People

The most addressed element – dependency on people or personnel – is divided into four groups. The first group that presented barriers to the implementation of AM was lack of management support. If full support from management was not forthcoming, Agile implementations were reported as either more painful and challenging[S1,S11] or less effective, due to the many problems and difficulties that arose[S23]. In the worst case, the implementation was terminated[S11]. In one case, the developers felt insecure and threatened in their Agile adoption status because a change in key management personnel led to suspicion against them and so to work impairment[S9]. In one case

Table 4. Factors that imposed limitations in Agile adoption

Category of limitations in Agile Implementation	Frequency	Literature support
High dependency on people/personnel	10	Moe et al., 2008; Rottier and Rodrigues, 2008; Thomas, 2008; Evans, 2008;
• Top/Senior management		Shatil et al., 2010; Hajjdiab and Taleb, 2011; Bass, 2012; Asnawi et al., 2012; Kaiser and Royse, 2011; Jakobsen and Poppendieck, 2011
• Developer (experienced /high-skilled)	8	Xu and Lippert, 2007; Moe et al., 2008; Smith and King, 2008; Babar, 2009; Savolainen et al., 2010; Smits and Rilliet, 2011; Kaiser and Royse, 2011; Middleton and Joyce, 2012
• Others (SCRUM Master, Product Owner, etc,)	8	Korkala and Abrahamsson, 2007; Moe et al., 2008; Srinivasan and Lundqvist, 2009; Lehto and Rautiainen, 2009; Moe et al., 2009; Hajjdiab and Taleb, 2011; Kaiser and Royse, 2011; Jakobsen and Poppendieck, 2011
• Customer	4	Korkala and Abrahamsson, 2007; Cao and Ramesh, 2008; Jakobsen and Poppendieck, 2011; Cagle, 2012
Organisational dependency	7	Thomas, 2008; Evans, 2008; Srinivasan and Lundqvist, 2009; Greening, 2010; Kaiser and Royse, 2011; Bass, 2012; Asnawi et al., 2012
High impact on the organisational structure and culture	6	Srinivasan and Lundqvist, 2009; Moe et al., 2009; Hajjdiab and Taleb, 2011; Middleton and Joyce, 2012; Asnawi et al., 2012; Thomas and Baker, 2008
Rely heavily on communication	6	Korkala and Abrahamsson, 2007; Moe et al., 2008; Cao and Ramesh, 2008; Lehto and Rautiainen, 2009; Miller and Haddad, 2012; Asnawi et al., 2012
Demand for stable resources	6	Moe et al., 2008; Moe et al., 2009; Hajjdiab and Taleb, 2011; Kaiser and Royse, 2011; Middleton and Joyce, 2012; Miller and Haddad, 2012
Agile adoption is time-consuming	6	Moe et al., 2008; Moe et al., 2009; Smits and Rilliet, 2011; Hajjdiab and Taleb, 2011; Kaiser and Royse, 2011; Asnawi et al., 2012
Demand for high transformation in leadership	5	Thomas and Baker, 2008; Evans, 2008; Moe et al., 2009; Smits and Rilliet, 2011; Kaiser and Royse, 2011
Rely on collocation	5	Xu and Lippert, 2007; Smith and King, 2008; Thomas, 2008; Jakobsen and Poppendieck, 2011; Asnawi et al., 2012
Lack of specific guidelines in the practical applications	5	Rottier and Rodrigues, 2008; Srinivasan and Lundqvist, 2009; Lehto and Rautiainen, 2009; Moe et al., 2009; Asnawi et al., 2012
Lacking in tools	5	Srinivasan and Lundqvist, 2009; Lehto and Rautiainen, 2009; Savolainen et al., 2010; Smits and Rilliet, 2011; Azizyan et al., 2011
Lacking in concrete measurement/ measurement mechanism	4	Smith and King, 2008; Thomas, 2008; Shatil et al., 2010; Greening, 2010
Lacking in architectural-related practices	4	Rottier and Rodrigues, 2008; Babar, 2009; Savolainen et al., 2010; Cagle, 2012
Risk in minimal documentation	4	Xu and Lippert, 2007; Cao and Ramesh, 2008; Smits and Rilliet, 2011; Cagle, 2012
No focus for maintenance program	4	Xu and Lippert, 2007; Hanssen et al., 2009; Babar, 2009; Savolainen et al., 2010
Limited support for large and complex system	4	Xu and Lippert, 2007; Hanssen et al., 2009; Beckhaus et al., 2009; Bass, 2012
Inadequate for non-software disciplines/components	3	Shatil et al., 2010; Savolainen et al., 2010; Smits and Rilliet, 2011
Demand for trust	3	Moe et al., 2008; Cao and Ramesh, 2008; Asnawi et al., 2012
Demand for continuous learning	3	Srinivasan and Lundqvist, 2009; Moe et al., 2009; Hajjdiab and Taleb, 2011
Limitations in Product Backlog	2	Moe et al., 2009; Savolainen et al., 2010
Limitations in User Stories	2	Babar, 2009; Savolainen et al., 2010
Limitations in Refactoring	2	Hanssen et al., 2009; Chookittikul et al., 2011
Scarce in Requirements Engineering (RE)	1	Cao and Ramesh, 2008
Traded-in quality	1	Hanssen et al., 2009

study that reported success in the implementation of SCRUM, strong management support has been gained from the beginning, and beyond that, the company had been practising SCRUM for five years[S13]. On the other hand, another company was reported as clearly failing, with lack of management support being the main cause[S11]. In this case, management approval was said to be fundamental in realising AM.

The need for experienced and high-skilled developers was the second most important factor in the people dependency constraint. When adopting AM, using merely product backlog in place of traditional Requirements Engineering (RE) practices requires vast expertise to handle the huge gap between the user requirements and coding[S22]. Besides, the lack of documentation[S29], the emphasis on working autonomously (both individually and as a team) through the self-organising approach[S19], having no design work prior to applying the User Stories technique for requirements elicitation[S3] and doing things in a speedy way without adequate experience – all lead to the so-called 'technical debt'. This is a term utilised by Kaiser and Royse[S14] that uses financial debt as an analogy to code issues where; 'when a developer cuts a corner (whether they are implicitly asked to do so or not) it is potentially something that the company will pay for down the road. That is because it will generally cost more (sometimes much more) to resolve it later'[S14]. All of these conditions were addressed as factors contributing to the need for highly skilled and experienced developers or project teams in exercising AM. The fact that Agile approaches require skilled personnel is acknowledged in Smith and King's[S24] experience report on a project that adopted the XP method: 'a clear benefit was the high level of expertise of candidates that applied for positions throughout the project.

Subsequent dependency concern was evident in other personnel, such as the Agile/SCRUM master and the Product Owner (PO). To start with, a clear specification of each and every person's role and tasks for the Agile project team is a must, according to Jakobsen and Poppendieck[S13], who base their view on their multiple case study findings. In one case, a clear blueprint of the PO's roles (to the extent of laying down "how the team collaborated with the PO" is recognised as having been a key factor in successful project sprints[S13]. Nevertheless, to date no proper guidelines are available for either the PO's role and job description, the Agile/SCRUM master or the other main roles in an Agile project. Few articles discuss the PO's role, regardless of the clear need for such discussion[S16]. To conclude, the applications of the positions of both the PO and the Agile/SCRUM master in the studied literatures are inconsistent, and disappointment is reported in regard to them[S11,S14,S16,S19,S20,S26]. They were either inexperienced, inactive, rarely available, unavailable, or inefficient. Those who want to adopt AM need to invest extra initiative and effort in carefully designing the main roles and job descriptions, based on their project requirements and environment.

Last but not least, Agile practices depend highly on intensive communication with the customer, since customer collaboration is one of the four core values of Agile methodologies. As a result, when there is a lack of trust on the customer's side or the customer is unavailable, the relevant information and feedback will be in scarce[S7]. The "information shortage" issue is also identified in Korkala and Abrahamsson's[S15] report, in which the lack of a well-defined customer is said to have caused serious problems. In their case (a distributed Agile project), the customer became intermediate between two remotely located teams, but exercised the responsibility passively and failed to transmit important information between the two teams. Jakobsen and Poppendieck[S13] mention that, 'many projects struggling with clarifying features in collaboration with the customer. Clarifications from the customer were late, leading to a decrease in flow, which we know causes schedule and cost overrun'. In the case, too much was expected of the customer, outstripping his ability and readiness [S13].

4.2.2 Organisational Dependency

Srinivasan and Lundqvist[S26] reported how, given a lot of investment had been made on a certain tool, a company mandated its use to support Agile processes, even though it was inappropriate. Meanwhile, organisational restructures had obstructed communication and collaboration between developers and testers, as well as with QA team when they were placed separately from each other[S27]. According to Bass[S4], organisational boundaries such as enterprise policies that separate the development team from the team for testing and release activity make Agile technique like Test-Driven Development (TDD) impossible. Policies of this kind disallow the adoption of testing process like TDD, besides being impractical

to incorporate in a short release cycle, due to the longer time taken for testing activity. 'Organizational resistance may be the main barrier to other organizations trying it, because top executives and engineers must be willing to give it a serious try'[S10]. This refers to a SCRUM practice extension initiative at the executive level. To truly gain people's commitment to the changes brought about by the values of AM, the effort must come from the highest organisational level[S1]. Lalsing et al[11] conclude: "it is imperative that before any organisation decides to adopt an AM, it needs to assess whether the company culture, operating structure, business processes and projects are suited for the use of an Agile Project Management Methodology".

4.2.3 High Impact on Organisational Structure and Culture

To adopt AM, there will be a need for changes not only in the working style, but also in the organisational structure and culture whenever necessary. For example, Thomas and Baker[S28] state that there is 'an inherent conflict' between AM and organisations operated under legacy processes, mindsets and cultures, while Middleton and Joyce[S17] point out the drawback that Lean methods might conflict with current corporate standards. Among the obstacles specified, 'Lean does not work well with targets, milestones, Gantt charts and traffic-light reporting methods' (the artifacts/requirements demanded in organisation with heavy plan-driven processes). This is quite apart from the need for changes in the job roles and responsibilities such as the manager being a facilitator[S17] instead of a planner and controller[13]. To conclude, organisational changes are necessary in the Agile adoption process[S11,S20], especially for traditional corporate governance[S28]. Nevertheless, culture and mindset among the people within organisations are recognised as the most difficult aspects to change[S25].

4.3 Summary of this Review

Firstly, AM limitations from the final 29 relevant papers were compiled and grouped through detailed analysis made throughout the SLR study. Meanwhile, discoveries and inputs from other literature readings (i.e.)[14] contributed to the refinement of the limitation items, as finalised in Table 4. Nonetheless, instead of classifying the limitations based on project sizes (small, medium and big) and/or project types (non-distributed and distributed), they are generally itemised.

Secondly, two factors – organisational dependency, and high impact on organisational structure and culture – are separated from each other. While organisational dependency refers to constraints posed by a decision at the organisational level, the latter factor signifies required organisational changes that may arouse strong opposition and challenges. To conclude, these two factors are mutually related and can be associated back to the people factor.

To compare, this pilot SLR finding aligns with several earlier research results on the importance of high commitment and support from the upper level of management for better implementations of AM. For instance, Young and Jordan[17], reporting multiple case studies on projects that 'ranged from complete failure to complete success', show that Top Management Support (TMS) is the most critical determinant for project success or failure. In his worldwide online survey, Livermore[12] found that 'a number of the factors that impact the implementation of an Agile development methodology are completely under the control of management'. Lastly, a recent study on Agile deployment in three companies identified clear management vision and support as very significant and thus provided recommendations for management on effective Agile deployment plan[15].

4.4 Limitations of this Review

This SLR only reviewed papers in IEEE (one e-database only). Therefore, this review study can be regarded as a pilot SLR to identify and prioritise the factors limiting Agile implementations in the software industry. Since the SLR inclusion criteria are broad, including case studies, empirical studies, experience reports, surveys, and expert opinion in all project types (small, medium and large) in both distributed and non-distributed Agile project management and development, using only one e-database has taken considerable time. This is because the whole SLR process is done by means of a profound analysis on the part of a single reviewer.

This SLR study does not specifically include the obstacles to implementing AM in distributed projects. We found that other than technology restrictions, the relevant papers do not distinctly stress the limitations concerning distributed environments. In spite of that, the shortcomings in implementing AM in such project settings are an important and huge area to be focused on its own.

5. Conclusion and Further Work

From the 29 most relevant papers identified in the IEEE database, a high dependency on people/personnel was found to be the most mentioned factor imposing limitation to Agile implementations in the software industry. People dependency encompasses management, developers, customers, and others such as Agile master and product owner. Followed by organisational dependency as the second limiting factor and, high impact on the organisational structure and culture as the third limiting factor. The SLR demonstrates that the first three limiting factors (those with the highest frequency) originate mainly from the people factor, in which management plays the most important role. While the rest of the limitations suggest that most of them can also be strongly associated back to the people factor (management), they are not elucidated in this paper.

Our further study indicates that there are a lot of misconceptions with the Agile methodology. For instance, the addressed issue of 'lack of specific guidelines in the practical applications' was found as extraneous since AM are meant to be generative, not prescriptive[21]. Specific processes or guidelines would inhibit the purpose of Agile methodology which is; to encourage long lasting growth and improvement in people through a healthy working environment. Indeed, for a software development where the task structure is of complex and non-routine, 'the leader may not be able to come up with specific guidelines. Operating procedures will have to be more flexible, and the leader will have to assume a greater role in guiding the process'[19]. It has been widely emphasized that the role of leaders or managers in Agile is facilitating. To conclude, the flexibility and generality of AM left them open to various interpretations, particularly when the underlying purpose behind the concept of agile is not well understood by the agile adopters[5,18,20].

Therefore, further research is necessary to identify the kind of apposite guidelines for management (especially upper level managers) on their actual role in the practical implementation of AM.

6. Acknowledgement

This research was supported financially by the Ministry of Higher Education Malaysia (FRGS- FP068-2015A).

Table 5. Related papers by year

Appendix

The 29 relevant papers:

S1. Asnawi AL, Gravell AM, Wills GB. Emergence of Agile Methods: Perceptions from software practitioners in Malaysia. AGILE India; 2012 Feb 17-19; Bengaluru, India: IEEE; 2012. p. 30-39.

S2. Azizyan G, Magarian MK, Kajko-Matsson M. Survey of Agile tool usage and needs. Agile Conference; 2011 Aug 7-13; Salt Lake City, UT. USA: IEEE; 2011. p. 29-38.

S3. Babar MA. An exploratory study of architectural practices and challenges in using Agile software development approaches. European Conference on Software Architecture; 2009 Sep 14-17; Cambridge. UK: IEEE; 2009. p. 81-90.

S4. Bass JM. Influences on Agile practice tailoring in enterprise software development. AGILE India; 2012 Feb 17-19; Bengaluru, India: IEEE; 2012. p.1-9.

S5. Beckhaus A, Karg LM, Hanselmann G. Applicability of software reliability growth modeling in the quality assurance phase of a large business software vendor. Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference; 2009 Jul 20-24; Seattle, WA. USA: IEEE; 2009. p. 209-15.

S6. Cagle R. Enterprise Architecture facilitates adopting Agile development methodologies into a DoD acquisition. IEEE International Systems Conference (SysCon); 2012 Mar 19-22; Vancouver, BC. Canada: IEEE; 2012. p. 1-5.

S7. Cao L, Ramesh B. Agile requirements engineering practices: An empirical study. IEEE Softw. 2008 Jan-Feb;25(1):60-67.

S8. Chookittikul W, Kourik JL, Maher PE. Reducing the gap between academia and industry: The case for Agile Methods in Thailand. Eighth International Conference on Information Technology: New Generations (ITNG); 2011 Apr 11-13; Las Vegas, NV. USA: IEEE; 2011. p. 239-244.

S9. Evans M. The FrAgile Organisation. AGILE Conference; 2008 Aug 4-8; Toronto, ON. Canada: IEEE; 2008. p. 181-5.

S10. Greening DR. Enterprise Scrum: Scaling Scrum to the Executive Level. 43rd Hawaii International Conference on System Sciences (HICSS); 2010 Jan 5-8; Honolulu, HI. USA: IEEE; 2010. p. 1-10.

S11. Hajjdiab H, Taleb AS. Agile adoption experience: A case study in the U.A.E. 2nd International Conference on Software Engineering and Service Science (ICSESS); 2011 Jul 15-17; Beijing. China: IEEE; 2011. p. 31-34.

S12. Hanssen GK, Yamashita AF, Conradi R, Moonen L. Maintenance and agile development: Challenges, opportunities and future directions. IEEE International Conference on Software Maintenance (ICSM); 2009 Sep 20-26; Edmonton, AB. Canada: IEEE; 2009. p. 487-90.

S13. Jakobsen CR, Poppendieck T. Lean as a Scrum Troubleshooter. Agile Conference; 2011 Aug 7-13; Salt Lake City, UT. USA: IEEE; 2011. p. 168-174.

S14. Kaiser M, Royse G. Selling the investment to pay down technical debt: The code Christmas tree. Agile Conference; 2011 Aug 7-13; Salt Lake City, UT. USA: IEEE; 2011. p. 175-

180.
S15. Korkala M, Abrahamsson P. Communication in distributed Agile development: A case study. 33rd EUROMICRO Conference on Software Engineering and Advanced Applications; 2007 Aug 28-31; Lubeck. Germany: IEEE; 2007. p. 203-210.
S16. Lehto I, Rautiainen K. Software development governance challenges of a middle-sized company in Agile transition. ICSE Workshop on Software Development Governance (SDG '09); 2009 May 17-17; Vancouver, BC. Canada: IEEE; 2009. p. 36-39.
S17. Middleton P, Joyce D. Lean software management: BBC Worldwide case study. IEEE Trans Eng Manag. 2012 Feb;59(1):20-32.
S18. Miller JR, Haddad HM. Challenges faced while simultaneously implementing CMMI and Scrum: A case study in the tax preparation software industry. Ninth International Conference on Information Technology: New Generations (ITNG); 2012 Apr 16-18; Las Vegas, NV. USA: IEEE; 2012. p. 314-318.
S19. Moe NB, Dingsoyr T, Dybå T. Understanding self-organizing teams in Agile software development. 19th Australian Conference on Software Engineering (ASWEC); 2008 Mar 26-28; Perth, WA. USA: IEEE; 2008. p. 76-85.
S20. Moe NB, Dingsyr T, Kvangardsnes O. Understanding shared leadership in Agile development: A case study. 42nd Hawaii International Conference on System Sciences (HICSS '09); 2009 Jan 5-8; Big Island, HI. Australia: IEEE; 2009. p. 1-10.
S21. Rottier PA, Rodrigues V. Agile development in a medical device company. AGILE Conference; 2008 Aug 4-8; Toronto, ON. Canada: IEEE; 2008. p. 218-23.
S22. Savolainen J, Kuusela J, Vilavaara A. Transition to Agile development: Rediscovery of Important Requirements engineering practices. 18th IEEE International Requirements Engineering Conference (RE); 2010 Sep 27- Oct. 1; Sydney, NSW. Australia: IEEE; 2010. p. 289-94.
S23. Shatil A, Hazzan O, Dubinsky Y. Agility in a large-scale system engineering project: A case-study of an advanced communication system project. IEEE International Conference on Software Science, Technology and Engineering (SWSTE); 2010 Jun 15-16; Herzlia, Israel. Israel: IEEE; 2010. p. 47-54.
S24. Smith C, King P. Agile project experiences: The story of Three Little Pigs. AGILE Conference; 2008 Aug 4-8; Toronto, ON. Canada: IEEE; 2008. p. 378-83.
S25. Smits H, Rilliet K. Agile experience report: Transition and complexity at Cisco Voice Technology Group. AGILE Conference; 2011 Aug 7-13; Salt Lake City, UT. USA: IEEE; 2011. p. 274-8.
S26. Srinivasan J, Lundqvist K. Using Agile methods in software product development: A case study. Sixth International Conference on Information Technology: New Generations (ITNG '09); 2009 Apr 27-29; Las Vegas, NV. USA: IEEE; 2009. p. 1415-20.
S27. Thomas J. Introducing Agile development practices from the middle. 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS); 2008 Mar 31-April 4; Belfast. Northern Ireland: IEEE; 2008. p. 401-7.
S28. Thomas JC, Baker SW. Establishing an Agile portfolio to align IT investments with business needs. AGILE Conference; 2008 Aug. 4-8; Toronto, Ont. Canada: IEEE; 2008. p. 252-8.
S29. Xu J, Lippert D. Lesson learned in managing IT departments. Portland International Center for Management of Engineering and Technology; 2007 Aug 5-9; Portland, OR. USA: IEEE; 2007. p. 2107-15.

8. References

1. Abrahamsson P, Warsta J, Siponen MT, Ronkainen J. New directions on Agile methods: A comparative analysis. IEEE Proceedings of the 25th International Conference on Software Engineering; USA. 2003 May 3-10. 2003. p. 244-54.
2. Asnawi AL, Gravell AM, Wills GB. Empirical investigation on Agile methods usage: Issues identified from early adopters in Malaysia. Proceedings of the 12th International Conference (XP 2011); Madrid, Spain. 2011 May 10-13. p. 192-207.
3. Bustard D. Beyond mainstream adoption: From Agile software development to Agile organizational change. IEEE 19th International Conference and Workshops on Engineering of Computer based Systems; Novi Sad, Serbia. 2012 Apr 11-13. p. 90-7.
4. Causevic A, Sundmark D, Punnekkat S. Factors limiting industrial adoption of test driven development: A systematic review. IEEE 4th International Conference on Software Testing, Verification and Validation; Berlin. Germany. 2011 Mar 21-25. p. 337-46.
5. Conboy K, Fitzgerald B. Toward a conceptual framework for Agile methods: A study of agility in different disciplines. Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research; New York, NY. USA. 2004. p. 37–44.
6. Dingsoyr T, Dyba T. Empirical studies of Agile software development: A systematic review. Inform Software Tech. 2008; 50(9-10):833-59.
7. Fowler M, Highsmith J. The Agile manifesto. 2001. Available from: www.pmp-projects.org/Agile-Manifesto.pdf
8. Fowler M. The new methodology. 2005. Available from: http://www.martinfowler.com/articles/newMethodology.html
9. Jalali S, Wohlin C. Agile practices in global software engineering: A systematic map. 5th IEEE International Conference on Global Software Engineering; Princeton, NJ. USA. 2010 Aug 23-26. p. 45-54.
10. Kitchenham BA, Charters S. Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report. Keele University and University of Durham (UK); 2007.

11. Lalsing V, Kishnah S, Pudaruth S. People factors in Agile software development and project management. Int J Softw Engin Appl. 2012; 3(1):117-37.
12. Livermore JA. Factors that significantly impact the implementations of an Agile software deployment methodology. J Softw. 2008; 3(4):31-6.
13. Nerur S, Mahapatra RK, Mangalaraj G. Challenges of migrating to Agile methodologies. Commun ACM. 2005; 48(5):73-8.
14. Neves FT, Correia AMR, Rosa VN, de Castro Neto M. Knowledge creation and sharing in software development teams using Agile methodologies: Key insights affecting their adoption. IEEE 6th Iberian Conference on Information Systems and Technologies; Chaves. Portugal. 2011 Jun 15-18. p. 1-6.
15. Pikkarainen M, Salo O, Kuusela R, Abrahamsson P. Strengths and barriers behind the successful Agile deployment: Insights from the three software intensive companies in Finland. Empir Softw Engin. 2012; 17(6):675-702.
16. West D, Grant T. Agile development: Mainstream adoption has changed Agility Forrester Research. 2010. Available from:http://wales.heacademy.ac.uk/assets/bmaf/documents/events/Events_2011/agile_development_mainstream_adoption_ has_changed_agility.pdf
17. Young R, Jordon E. Top management support: Mantra or necessity? Int J Proj Manag. 2008; 26(7):713-25.
18. Yu X, Petter S. Understanding Agile software development practices using shared mental models theory. Inform Software Tech. 2014; 56(8):911-21.
19. Beekun RI, Badawi J. Leadership: An Islamic perspective. USA: Amana Publications; 1999.
20. Conboy K. Agility from first principles: Reconstructing the concept of agility in information systems development. Inf Syst Res. 2009 Sep; 20(3):329-54.
21. Highsmith J. Agile project management. Boston, MA: Addison-Wesley; 2010.