

Large Scale Agile Adoption Model from Management Perspective

Muhammad Faisal
Abrar
Department of
Computer Science and
IT,
Qurtuba University,
Pakistan

Sajid Rehman
Department of
Computer Science and
IT,
Qurtuba University,
Pakistan

Nasir Rashid
Department of
Computer Science & I.T,
University of
Malakand, Pakistan

Sikandar Ali
Department of
Computer and
Software Technology,
University of Swat,
Pakistan

ABSTRACT

The field of software engineering is growing rapidly due to increased change of customers' requirements and progress in technologies. Agile software development (ASD) have substituted the conventional Software development methodologies to enhance the software productivity through short increments and increased user satisfaction. The purpose of this research is to pinpoint the factors (success factors and risk factors) in adopting agile methods for large scale software development team from management perspective. We have adopted systematic literature review (SLR) as a research methodology for the identification of these factors. The expected outcomes of this research will provide input to our proposed model Large Scale Agile Adoption Model (LSAAM) from management perspective. The LSAAM intends to assist the agile experts in managing large team for software development.

Keywords

Agile software development, success factors, large development team, Systematic literature review.

1. INTRODUCTION

Agile Methods were basically planned/devised for use in small, single-team projects (Boehm & Turner, 2005). Nevertheless, their overt and covert advantages / profits have made them appealing also outside this purview, and particularly scale development. In spite of the fact that they are more difficult to implement in larger projects (Dyba & Dingsoyr, 2009), vis a vis small projects, icon for agile development, larger ones are featured by the prerequisites for supplementary coordination. A specific problem in adopting agile to larger projects is how to entertain/cope with inter-team coordination. Large-scale agile involves extra apprehension in inter-coping with other organizational units, such as human resources, marketing and sales, and product management. Besides, large scale may make users and other stakeholders become far off from the development teams. In spite of these overt problems related to large-scale agile development, there is an industry tendency towards adopting agile methodologies in-the-large (DingsÅ,yr & Moe, 2014; Paasivaara, Behm, Lassenius, & Hallikainen, 2014; Paasivaara, Lassenius, HeikkilÅ, Dikert, & Engblom, 2013).

The State of Agile Survey that Version One (versionone, 2016) has been organizing annually since 2007, lately has asked a couple of questions associate with large scale as well, e.g. on scaling methods used, and guidelines for improvement with scaling agile. According to the most up to date survey (versionone, 2016), 62% of the almost 4000 respondents had

more than hundred people in their software organization and 43% of all the replients participated in development organizations where more than half of the team has agile experts. Certainly, the sample of this study is shrunk to a chosen subset of companies and countries (of the almost 4000 replies to the latest survey 65% were from North America and 26% from Europe). Nevertheless, this shows that there appears to survive a large number of companies that have adopted or aims to adopt agile methods into practice in large-scale environment (versionone, 2016).

"Agile Software Development" is the aggregate of iterative and incremental software engineering methods that are supported on an "Agile Philosophy" snapped in the "Agile Manifesto" (Fowler & Highsmith, 2001). While mostly recollecting previously well-known good software development practices, the agile movement can be contemplated as an substitute for so called traditional software development methodologies. While a traditional approach stresses/emphasizes on candid planning and rigorous management of change, agile methods were designed to admit, and wisely manage change (Cockburn & Highsmith, 2001; Fowler & Highsmith, 2001). Agile methods have been both condemned and supported and research has shown that adjusting/supporting/accepting change may be a factor in both success and failure (Boehm, 2002). It has been exhibited/brought to light that agile approaches have developed trust of both customers and developers, but on the other hand there is proof that agile approach may not be well fit for large tasks(Dyba & Dingsoyr, 2009). A proposed solution is that each organization looks for its own balance of agile and plan driven methods (Boehm, 2002).

Two of the most common Agile Approaches are Extreme Programming (XP) and Scrum (Hamed & Abushama, 2013). The problem of presenting agile methods enhances with the development team size (DybÅ & DingsÅ,yr, 2008). The problem is partially connected with size creating higher organizational stagnation which mitigates organizational change (Livermore, 2008). Agile development is not edified on the use of individual tools or practices, but rather on a universal way of thinking. Adopting agile time and again requires change of the whole organizational edifice (Chandra Misra, Kumar, & Kumar, 2010). One noticeable rift between small and large-scale adoptions is that larger organizations have more reliance between projects and teams. This elevates the desire for formal documentation, and thus mitigates agility (Lindvall et al., 2004). Moreover, to inter-team coordination, development teams have to interact with other organizational units, which are time and again non-agile in features. For example, human resources unit may require and need

individuals to have severely indicated roles in projects (Boehm & Turner, 2005), or a change control board may inhibit the use of continuous repackages (Lindvall et al., 2004). All units hit by the change to agility to require to be notified and discussed, and the agile process must be made familiar according to their requirements (Boehm & Turner, 2005; Cohn & Ford, 2003; Lindvall et al., 2004).

Agile methods also upset management and business concerned functions. A key tasks/test is that management has to move away from life-cycle models and towards iterative and feature centric models (Nerur, Mahapatra, & Mangalaraj, 2005), which need a change of approach. The emphasis/stress has to be adopted from long-term planning to shorter term project planning (Chandra Misra et al., 2010), as agile methods stress/urge that planning is only consequential/useful for the near future (Cohn & Ford, 2003). Nevertheless, the dearth of planning can be a matter of worry, because business and customer relationships time and again develop on long range plan of action. Coordinating/synthesizing operation with shorter term planning needs socializing and aware of stakeholders and scrutinizing contracting practices (Boehm & Turner, 2005).

While the research literature includes a bunch of experience reports, and some case studies on large-scale agile adoption from management perspectives, a systematic overview and fusion of this nurturing and developing body of research is still not existing. Freudenberg and Sharp (Freudenberg & Sharp, 2010) inquired the industrial practitioners at the XP2010 (xp, 2010) to erect a back log of topics, they think should be studied. As the leading hot research question, the practitioners voted "Agile and large development teams". Nevertheless, out of the top ten items three emphasized on distributed agile development, which is related, particularly, for the larger organizations as they are time and again geographically disseminated. Research on agile software development is amassing and developing, and has provided a basis for executing systematic literature reviews (DybbÅyr & DingsÅyr, 2008; Jalali & Wohlin, 2012; Kaisti et al., 2013), the extent and premises of large-scale agile development from management perspectives has not yet been attempted through secondary studies. In this paper, we start filling in the gap with presenting the systematic literature review of large-scale agile software development team from management perspectives.

2. LITERATURE REVIEW

A through literature search (DingsÅyr, FÅgri, & Itkonen, 2014) ascertained previous explanations of what large-scale agile is. Size had been considered in expression of size in personnel or teams, project budget, code base size, and project spell. The examples of cases that were called "large-scale" consisted 40 people and 7 teams (Paasivaara, Durasiewicz, & Lassenius, 2008), project cost of over 10 million GBP and a team size of over 50 people (Berger & BeynonDavies, 2009), a code base size of over 5 million lines of code (Petersen & Wohlin, 2010), and a project time of 2 years with a project range of 60-80 traits/characteristics (Bjarnason, Wnuk, & Regnell, 2011). Based on their findings

DingsÅyr et al. (DingsÅyr et al., 2014) wound up measuring large-scale by the number of collaborating and coordinating teams: they categorized as large-scale 2-9 collaborating teams and as very large-scale over ten collaborating teams. We founded a number of additional studies discussing large-scale ASDM and their interpretations of large-scale. All of these discussed to the number of people involved. In early work on

agile, Fowler think about the Crystal methodology to be appropriate for up to 50 people (Fowler, 2000). The same number has been testified as seen by practitioners and researchers as the size of the largest organization appropriate for agile (Williams, 2010). Other studies have referred to agile projects including up to 50 people as small (Koehnemann & Coats, 2009), and considered a development project large if it had a staff between 50 and 100 people, including all project personnel (Elshamy & Elssamadisy, 2006). The largest numbers were 300 people across 3 sites (Moore & Spens, 2008).

Participants of the XP2014 large-scale agile workshop gave very varying definitions for large-scale agile development (DingsÅyr & Moe, 2014), showing that what is seen as large-scale depends very much on the context and the person defining it. Based on these findings, we defined large-scale to denote software development organizations with 50 or more people or at least six teams. All people do not need to be developers, but must belong to the same software development organization developing a common product or project, and thus have a need to collaborate. For instance, Scrum masters and software architects are counted when assessing the organizational size. As some studies present the number of teams rather than the number of people, we correspondingly defined large-scale to denote development efforts involving at least six teams. Having six teams with an average size of six to seven people plus a number of supporting staff can reasonably be considered to form an organization of 50 people. In this definition, we include both companies that as a whole focus on software development, as well as the parts of larger (non-software focused) organizations that develop software, e.g. in-house software development units of large non-software corporations.

3. RESEARCH FOCUS

The motivation of this research study comes from exploring how agile methodologies are adopted for large scale software development team from management perspective. Keeping in view the above mentioned importance of agile approaches for large scale development, there is intensive needs to erect a model that will assist the project managers in such make possible the implementation of agile methods in such environments.

With the above mentioned aim in mind, we have formulated the following research questions (RQs).

RQ1. What are the success factors, as identified in the literature, in adopting agile methods for large scale software development teams from management perspectives?

RQ1. What are the risk factors, as identified in the literature, in adopting agile methods for large scale software development teams from management perspectives?

RQ3. What are the success factors and risk factors, as identified in the real-world practice, in adopting agile methods for large scale software development teams from management perspective?

4. RESEARCH METHODOLOGY

We have adopted systematic literature review (SLR) (Keele, 2007; Staples & Niazi, 2007) as a research methodology for the development of the LSAAM. Our model development consists of the following three phases.

Phase#1: In this phase data is collected.

Phase#2: An empirical study will be conducted for validation of the findings from SLR and to identify the practices for the identified factors.

Phase#3: For evaluation of the LSAAM, case study will be conducted in software industries.

On the basis of relevant research question, SLR helps to find, explore and evaluate relevant data from the primary studies (Keele, 2007; Staples & Niazi, 2007). SLR is a thorough and unbiased method for collecting; interpreting and evaluating all published literature related to a particular topic, research question or phenomenon of interest (Staples & Niazi, 2007). Different types of technical, methodological and scientific value exist for SLR which differentiates it from ordinary literature surveys. SLR have three phases, planning is the first phase, where on the basis of the research questions a protocol is developed, implementation is the second phase, where questionnaire survey, case study is conducted and reporting is the third and last phase where the task related to preparation and publication of final report is performed (Staples & Niazi, 2007). Since SLR is reversible and unbiased methodology, therefore it will provide a high level of validity in decision about, selecting, evaluating and summarizing all available literature on Agile Methodology. For successful implementation of the factors (CSFs and CRs) that have a positive or negative role in promoting the existing agile methodology for large scale development team, a questionnaire survey will be conducted with experts working in the IT Industry.

The main objective of the survey is:

- Validation of the SLR findings
- Identification of any new factors (CSFs and CRFs) not previously identified through SLR.
- Identification of practices for addressing the identified CSFs and CRFs.

LSAAM will be evaluated through case studies in software industry to evaluate the effectiveness and applicability of LSAAM.

5. DEVELOPMENT OF LSAAM

The below mentioned two criteria will be used for the development and assessment of LSAAM.

A. User satisfaction: This criteria emphasizes the satisfaction of end user from the result of LSAAM. Project managers should be able to use the LSAAM without any confusion or ambiguity to achieve the specified goals according to their requirements and expectations.

B. Ease of use: This criterion emphasizes the structure easiness of the LSAAM. It states that the structure of the LSAAM should be flexible and self-understandable because organizations do not adopt complex models and standards which require resources, training and effort. Data collection and analysis are conducted during stage 2.

Rationalization and structuring of results are performed in stage 3.

Development of LSAAM based on the empirical findings is carried out in stage 4.

Evaluation of the LSAAM via case studies is performed in final stage i.e stage 5, as shown in Figure 1.

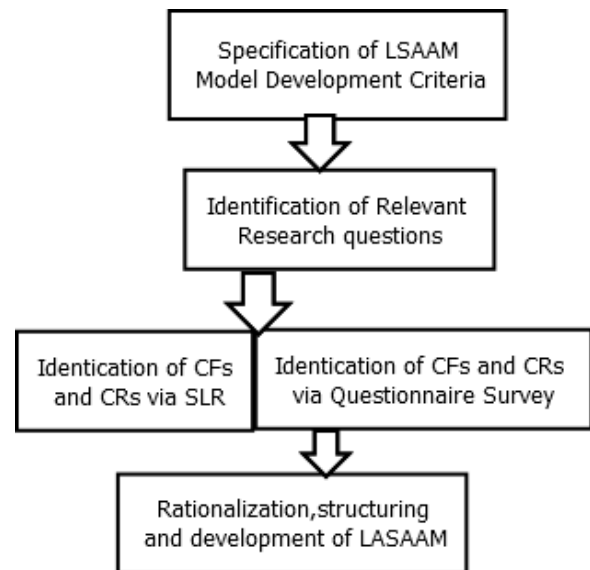


Figure 1: LSAAM Development Cycle Activities

5.1 LSAAM Structure

We will build the structure of the LSAAM on the following three dimensions, as shown in Figure 2.

- LSAAM levels
- Factors (CSFs, CRFs) in each level
- Practices for the implementation of agile methods for large scale development team from management perspective. Each Level of the LSAAM will have different factors (CSFs and CRFs). Under each CSFs and CRFs different practices will be assigned. For an organization to achieve certain LSAAM level they should address each CSF and CRFs under that particular level. Planned structure of the LSAAM is shown in Figure 2. It also show the relationship between LSAAM levels, factors/risks and different practices used.

5.2 LSAAM Evaluation

To check the validity and robustness of LSAAM in industrial environment, case study will be conducted in at least three software industries. Case study is a powerful tool for the evaluation of any software development model in real world environment. Moreover, we will also obtain important real word information through cased studies (Keele, 2007; Staples & Niazi, 2007) . Three case studies will be enough to evaluate the effectiveness of LSAAM in Large Scale Development team.

To obtain feedback from the participant about the LSAAM, a focus group sessions will be arranged (Morgan, 1996). For the structure of the focus group sessions, the criteria will be ease of use and user satisfaction as described in section 5.1. The data from focus groups is mainly from the interaction between the members in the group. In comparison to individual interview, focus group session is more open. The existence of subjects in a group of peers allows focus group session to be more open about issues discussed than individual interviews (Morgan, 1996).

6. CONCLUSION

Despite the evidence that agile methods have been adopted in software development, its applications in large scale development teams from management perspective still have to gain momentum. This paper presents the development of

proposed model (LSAAM) that intends to assist the project managers in adopting agile methods for large scale development team from management perspective.

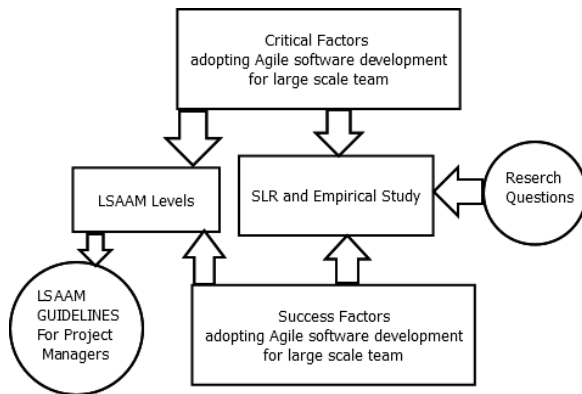


Figure 2: LSAAM Structure

7. REFERENCES

- [1] Berger, H., & Beynon Davies, P. (2009). The utility of rapid application development in large-scale, complex projects. *Information Systems Journal*, 19(6), 549-570.
- [2] Bjarnason, E., Wnuk, K., & Regnell, B. r. (2011). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. Paper presented at the Proceedings of the 1st Workshop on Agile Requirements Engineering.
- [3] Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.
- [4] Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE*, 22(5), 30-39.
- [5] Chandra Misra, S., Kumar, V., & Kumar, U. (2010). Identifying some critical changes required in adopting agile practices in traditional software development projects. *International Journal of Quality & Reliability Management*, 27(4), 451-474.
- [6] Cockburn, A., & Highsmith, J. (2001). *Software Management. Agile Software Development: The People Factor*. COMPUTER-LOS ALAMITOS-, 34(11), 131-135.
- [7] Cohn, M., & Ford, D. (2003). Introducing an agile process to an organization. *Computer*, 36(6), 74-78.
- [8] Dingsøyr, T., Fjærli, T. E., & Ikonen, J. (2014). What is large in large-scale? A taxonomy of scale for agile software development. Paper presented at the International Conference on Product-Focused Software Process Improvement.
- [9] Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development. Paper presented at the International Conference on Agile Software Development.
- [10] Dyba, T., & Dingsøyr, T. (2009). What do we know about agile software development? *IEEE software*, 26(5), 6-9.
- [11] Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9), 833-859.
- [12] Elshamy, A., & Elssamadisy, A. (2006). Divide after you conquer: an agile software development practice for large projects. Paper presented at the International Conference on Extreme Programming and Agile Processes in Software Engineering.
- [13] Fowler, M. (2000). Put Your Process on a Diet-As a reaction to cumbersome approaches to development, new methodologies have appeared. These methods attempt a compromise between no process and too much process. *Software Development*, 8(12), 32-39.
- [14] Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35.
- [15] Freudenberg, S., & Sharp, H. (2010). The top 10 burning research questions from practitioners. *Ieee Software*, 27(5), 8-9.
- [16] Hamed, A. M. M., & Abushama, H. (2013). Popular agile approaches in software development: Review and analysis. Paper presented at the Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on.
- [17] Jalali, S., & Wohlin, C. (2012). Global software engineering and agile practices: a systematic review. *Journal of software: Evolution and Process*, 24(6), 643-659.
- [18] Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Kärrläinen, K., Mäkilä, T., et al. (2013). Agile methods for embedded systems development-a literature review and a mapping study. *EURASIP Journal on Embedded Systems*, 2013(1), 1-16.
- [19] Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Technical report, Ver. 2.3 EBSE Technical Report*. EBSE.
- [20] Koehnemann, H., & Coats, M. (2009). Experiences applying agile practices to large systems. Paper presented at the Agile Conference, 2009. AGILE'09.
- [21] Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al. (2004). Agile software development in large organizations. *Computer*, 37(12), 26-34.
- [22] Livermore, J. A. (2008). Factors that significantly impact the implementation of an agile software development methodology. *Journal of Software*, 3(4), 31-36.
- [23] Moore, E., & Spens, J. (2008). Scaling agile: Finding your agile tribe. Paper presented at the Agile, 2008. AGILE'08. Conference.
- [24] Morgan, D. L. (1996). *Focus groups as qualitative research (Vol. 16)*: Sage publications.
- [25] Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
- [26] Paasivaara, M., Behm, B., Lassenius, C., & Hallikainen, M. (2014). Towards rapid releases in large-scale xaas development at ericsson: A case study. Paper presented at the 2014 IEEE 9th International Conference on Global Software Engineering.
- [27] Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). Using scrum in a globally distributed project: a case

- study. *Software Process: Improvement and Practice*, 13(6), 527-544.
- [28] Paasivaara, M., Lassenius, C., Heikkilä, V. T., Dikert, K., & Engblom, C. (2013). Integrating global sites into the lean and agile transformation at ericsson. Paper presented at the 2013 IEEE 8th International Conference on Global Software Engineering.
- [29] Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6), 654-693.
- [30] Staples, M., & Niazi, M. (2007). Experiences using systematic review guidelines. *Journal of Systems and Software*, 80(9), 1425-1437.
- [31] versionone. (2016). The VersionOne® Enterprise Agile Platform.
- [32] Williams, L. (2010). Agile software development methodologies and practices. *Advances in Computers*, 80, 1-44. xp. (2010). xp2010