

Agile Scalability for Large Scale Projects: Lessons Learned

Hina Saeeda^{1*}, Fahim Arif¹, Nasir Mehmood Minhas², Mammona Humayun²

¹ Department of Computer Software Engineering, MCS, National University of Sciences and Technology Islamabad Pakistan.

² Department of Computer Software Science, UIIT, Arid Agriculture University Pakistan.

* Corresponding author. Tel.:923335292948; email: hinasaeeda.mscs19@students.mcs.edu.pk.

Manuscript submitted December 10, 2014; accepted June 4, 2015.

doi: 10.17706/jsw.10.7.893-903

Abstract: In modern well known approaches, “agile” has emerged as the leading approach in software industry for the development of the software projects. With different innovative shapes agile is applicable for handling the issues regarding cost, time, continuously change environment and requirements. Agile has proved to be successful in the small and medium size project, however, it have several limitations when applied on large size projects. The aim of this study is to analyze agile approaches in detail, finding its success stories in small and medium size projects and highlighting its limitations for large size projects. This study will identify the current research problem of the agile scalability for large size projects by giving a detail literature review of the identified problem, and will synthesize the existing work for covering the identified problem in the agile scalability. Based on it, we can judge the limitations of agile scalability for large size projects and can think of some remedial approach for overcoming these limitations in future.

Key words: Agility, large scale projects, agile scalability, SCRUM, XP, DSDM, crystal, and disciplined agile delivery.

1. Introduction

In market different flavours of agility are available such as SCRUM, DSDM, CRYSTAL, XP, XP2 etc each having some unique and specific properties. While talking about agility it is very successful in the small and medium size project for implementing small and medium size projects with great results and success [1]. Agility is combination of the factors that leads the project towards success these factors make the project to have qualities like Modularity, Iterative, Time Bound, Parsimony, adoptions, Incrementality, Convergent, People-Oriented, Collaborations [2]. The agile qualities such as less documentation, pair programming, and high teamwork produce excellent results for the small and medium level projects large size projects are also using the approaches but with challenges [3]. In contrast, when we apply the agility on large size projects it does not provide desired results. When talking about scalability of agile two terms are in use “Scaling out” and “scaling up” Scaling up’ is concerned with using agile methods for developing large software systems that cannot be developed by a small team. Scaling out’ is concerned with how agile methods can be introduced across a large size projects with many years of software development experience [4]. It is not the true that 100 percent its application is going to fail but like the small level and medium level projects results, it does not show the same for large size projects. The agile approaches such as crystal – blue are in use for large size projects but show less agile properties [5]. Techniques of agility like SCRUM is applied for

the large size project but it has also some limitations [6]. The current research is focused on how to extend the agility towards regulated environment or large size projects and according to the researchers; it seems as the final frontier of agility approaches [7]. The discussion on ability of agile practices to scale to “large” software development efforts (more than 3 teams or 30 team members) has been widely debated in recent years [8]. Here the purpose of the research is to find the literature review on the agile scalability, identifying current work done for agile scalability and limitations faced by agility in large size projects.

2. Large Size Projects and Agile Limitations

Researchers reviewed agile processes to determine where and how to insert inventiveness techniques into them. They adopted Robert Sternberg’s definition of creativity as the ability to produce work that is both novel and appropriate. Therefore, they wanted to inject techniques that would generate novel and useful requirements. By useful, they are talking about requirements that, if satisfied by the software solution, would contribute positively to achieving one or more Project-specified goals [9]. In spite of agile software development being thought as Mainstream model from about 2010, the idea, and hence its strengths and restrictions, are still comparatively blurred. One approach to improve clarity is through analyzing real practice across a broad range of projects. Two such studies performed using detailed surveys of software development practice in companies in Northern Ireland in 2010 and 2012. The main aim of the first survey was to recognize the level of agile espousal in the area, while the second focused on understanding how agile principles and practices were growing and strengthening [10]. The implementation of agile development techniques has increased considerably in different software areas. This case study explains some of the benefits of the agile methods to the scientific-software development. Agile software practices are well suited to the investigative, iterative, and collaborative nature of scientific inquiry, especially when considering theory development through scientific experiments [11]. In the current cited paper the purpose of the research was to contribute with empirical proof on the impact of using agile principles and practices in large-scale, industrial software development. Research was paying attention on impacts within seven areas: Internal software documentation, Knowledge sharing, Project visibility, Pressure and stress, Coordination effectiveness, and Productivity [12]. The agile software development approach does not stop ensuring the security of software increments formed at the end of iterations. It proposes a technique for security guarantee of software increments and integrates security-engineering behavior into the agile software development [13]. Up to date software products solve more and more complex problems leading to growing complication of the software Products themselves. An uncontrolled growth of complication over a long time introduces a mixture of technical risks that have potential of jeopardizing the business of companies. There is an increasing need for frequently managing these risks, since in Lean/Agile software development the self-organized teams deliver small increments of software almost endlessly. The study developed a method and supporting measurement system for identifying the risky source code files and assessing the scale of the risk [14]. The extensive use of lean and agile development Methods show a fundamental move in how projects try to manage with complication and instability issue. In large-scale systems, the coordination of different people generally results in a team of teams’ system. The multi team systems point of view describes diverse conceptual policy types for interterm coordination. These types are explained with examples from a large enterprise software development organization [15]. This paper, answered the need to engage customers in the software development phase. Specifically saying this directs attention to large-scale software development where companies work hard to meet the specific requirements and requests of a large customer base. This paper is contributing in two-fold. First, suggest customer-specific teams (CST) as a way to build and maintain agile practices in the evolution phase of large-scale software development. Second, study insure about

customer-specific teams successful for improving 1) customer responsiveness, 2) customer satisfaction and 3) feature quality during software evolution [16]. A lightweight Release Framework can be very useful in scaling at the large set up level [17]. Regardless of the fact that lean and agile software development is popularly used recently, especially for larger-scale projects building complex products, the methodology leaves many architectural questions unsolved. For instance, agile techniques such as Extreme Programming suggests late architectural decisions and frequent remaking, while others suggest an “architectural runway” as infrastructure for a certain set of upcoming customer features. Software “product lines” consist of a set of software products that have a common set of characteristics. These product lines are developed from reusable main features incorporating changes in order to make customer-specific product variants. Therefore this research explores interoperability and complementarity of lean and agile methods in mixture with a software product line engineering methodology [18]. Issues like proper documentation and training in large size projects is also a major issue while applying the agile techniques. Factors like customer satisfaction, customer collaboration, customer commitment, decision time, corporate culture, personal characteristics, societal culture, training and learning have significant relationship with success of agile techniques in large size projects [19]. The current research issue is team and project size affect on agile process success. Researchers are focused on finding the relation between size of the project/company and agile techniques [20]. Agile application in large size project not produced complete agility results, even agile technique name scrum cannot be apply without making changes in it [21]. Scaling activities like scaling up agile is the problem according to the researchers the agile techniques such as scrum and others create problems when used for large size projects [22]. Agile practices such as code refracting, less documentation etc, are not feasible in large scale systems and here agile approaches have boundaries [23]. Agile software development is about feedback and change in the large size projects if applied agile techniques regular feedback and change will be needed [24]. Case studies are important to build proofs and making conclusions on software engineering tools and approaches. In this paper the researcher present a case study on an agile adoption for an organization in the United Arab Emirates (U.A.E). The institute follows the traditional Waterfall model and wants to use agile software engineering approach. Many challenges were faced by the agile teams which caused the use to fail. The main reasons for this ineffective adoption are lack of sufficient support and restricted financial and human resources [25]. Current research is identifying problem that ability of the agile methods to balance the execution cycle and the framework to be used successfully across large scale projects [26]. Peterson and Wohlin found that using agile and incremental techniques in large-scale software development leads to benefits in one part of the process, while creating issues in another part of the process. For example, using small and interacted teams’ increases control over the project, but leads to new problems at the management level where coordination of the projects has to take place [27]. They also stated the main challenges while using agile and incremental practices in large- scale software development as 1) challenges in regard to realize continuous testing, 2) increased maintenance effort with an increase in the number of releases, 3) management overhead due to the need for coordination between teams, 4) detailed dependencies are not discovered on a detailed level due to lack of focus on design, 5) long requirements engineering duration, due to complex decision processes in requirements engineering, 6) requirements priority lists are hard to create and maintain, 7) waiting times in the process, specifically in design waiting for requirements, 8) reduction of test coverage due to shortage of projects and lack of independent testing, 9) increased configuration management effort. According to some researchers existing agile principles do not support distributed development environment architecture. This paper focuses on the problems that arise when client is offshore and proposes an agile model for distributed development environment. 1. Poor documentation 2. Communication gap between Client and Developers are identified limitations in the

distributed projects while applied agile techniques [28]. The Scrum alternate of agile techniques is by far the most popular at Microsoft. The results also point out that developers are most concerned about scaling Agile to larger projects (greater than twenty members), attending too many meetings and the coordinating Agile and non-Agile teams. Many developers seemed problems with agile development techniques. Agile is facing problem when tried to scale, too many meetings, loss sight on big picture and lack of schedule [29]. The first challenge for large scale systems is the scope and complication. Secondly large size projects generally involve subcontracts with other businesses (sometimes competitors) for building or integrating substantial parts of the system. Third point is about Reporting governance, and fulfillment place extra load on large projects. Fourth point is that in many large organizations, a separate group “Quality Assurance”, (a.k.a. “Release Management”, “Configuration Management”) certifies software is suitable for release, while in agility this concept is missing [30]. Some authors have argued that ASD methods have limited applicability to large-scale projects because agile stories are not enough to get the complexities of up-front design. This research reports a 2.5-year field study of how an ASD team for a complicated software system modified the user story concept and the Scrum technique. The team wanted to create a convention for showing agile stories which could get the complexities of the system requirements without burdening the team with unwanted documentation [31]. Such as the factors multiple parties involvement, change impact on these parties, critical information about the mission and time pressure have great effect on the overall project. Scaling is an issue when applying Scrum in large enterprises. The large enterprise with collections of Scrum teams is far more complex than a single Scrum team and faces many scaling. Some researchers studied these agile scaling issues when adopting Scrum in the enterprise. These scaled agile studies identified a diverse set of issues, except for priority (related) issues which are mentioned by several studies. Petersen and Wohlin studied a case of Agile projects with Scrum characteristics, and identified issues with 1) creating and maintaining the priority list, 2) effort to setup and maintain a test basis that covers a sufficient part of the applications, 3) increased management overhead due to a high number of teams that requires coordination and communication and 4) increased configuration management effort due to an increased number of releases. The paper does not indicate whether the issues are identified in a Scrum chain setting. The issue with prioritization has also been identified by Lehto and Rautiainen, who studied Scrum issues in a mid-size software company, while the setting of the Scrum teams is ambiguous. In their study they conclude that prioritization of the high-level goals was unclear which impedes organizing and tracking development work. They also identified traceability issues from high level goals to detailed plans and a lack of information about progress which made it impossible to take corrective actions in time. Waardenburg and van Vliet studied two enterprises with Scrum teams. While the paper does not elucidate whether the teams are part of a chain, they also identified issues with prioritization, as a consequence of the lack of business involvement. Other identified issues are 1) requirements gathering problems, 2) limited business feedback, 3) communication problems and 4) dependencies between Definitions of Done. Other related work report experience with scaling. Saddington reports a success case about scaled product ownership in multiple co-dependent Scrum teams. They state that visibility and alignment of vision, goals, teams and workload were essential elements to the success of the project. Rautiainen et al. describe a company that implements an agile portfolio management structure to solve prioritization issues. Listing the projects in priority order on a single backlog creates visibility about ongoing projects, which benefits the coordination between Scrum teams. Even though agile principles aim to introduce flexibility the need for plans and structure remains.

3. Concept of Scaling in Agility

About scalability of the agile process in literature, two terms are used frequently i.e. scaling out and

scaling up. Scaling up is about agile methods for developing large software systems that cannot be developed by small teams. Scaling out is about how agile methods can be used across a large scale projects with several years of software development experience. Agility for Scale means scaling of agile methods for large size projects, this grouping focuses on disciplined agile delivery where one or more scaling factors are appropriate. The eight scaling factors are team size, geographical distribution, regulatory compliance, organizational complexity, technical complexity, organizational distribution, domain complexity, and enterprise discipline. All of these scaling factors are ranges, and not all of them will likely to be applicable to any given project, so you need to be flexible when scaling agile techniques to meet the needs of your specific condition. To deal with these scaling factors you will need to follow your disciplined agile delivery practices and in some cases accept a handful of new practices to address the additional risks that you face while scaling [4].

4. Efforts Already Made for Agile Scalability

A concrete framework for analyzing of the acceptance criteria for agile SDM to apply in different projects to apply has been recommended [32]. Agile scalability according to researchers can achieve by breaking the large projects into small projects [33]. In embedded it is recommended to keep in mind all the things relating to the adaptation of the agile scalability issues. According to researchers when applied the agile scalability to large size projects the main problem they faced is the lack of domain knowledge about the projects in detail see the recommended framework in Fig. 1 [34] in large setups the agile technique named totally integrated scrum methodology is applied where the teams are cross functional and the teams are distributed [35]. According to the recent techniques studied for the agile approaches scalability if applied the lean in combination with the scrum can help in the scalability of the agile approaches [21].

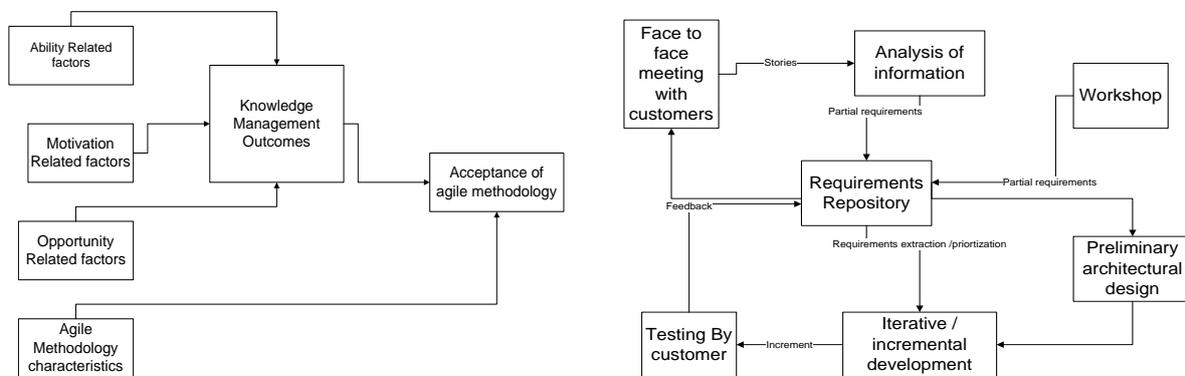


Fig. 1. Conceptual framework [34]. Fig. 2. Development process for complex software project [37].

In research it is clear that agile methods lack detail documentation but when talking about the scrum and XP it is proven that lack of documentation is not present but the less of documentation concept is present [36]. yahoo a large enterprise with a \$32 billion market cap and is the largest enterprise adopted agile methods when applied agile techniques in yahoo setup it showed the good result but it is also clear that patience is necessary thing to focus is on as it will be not an easy task to migrate from traditional water flow process to the agile at once [21]. A lot of the questions are still unanswered relating to the agile scalability and adoptability, researchers are advising to adopt some frameworks for scalability, but also identifying these proposed frameworks missed the agility properties [36] by adding different things to agile can make it flexible for large size projects such as dividing the large size teams into smaller size teams. Customers' coordination, communication and work in small projects and teams are hand able and easy to manage .Scrum provides central sharing for global teams in shape of product backlog and pair

programming is also applicable. Developers are also working for better testing and analyzing of the agile activities. All the done code is stored into the central code saving place, such as SourceSafe or Open Source the code done and stored is validated and analyzed. For each individual project there should be saving depository and a common place also to share all the projects done each project deliverable will be in small portions so that can be coordinated with other parts of the projects. At the end of every month the scrum will be entered into the final integration testing phase and all the done parts of project will be integrated with each other [37]. All the large scale projects faced a lot of problems in shape of the dynamic requirements change and the time to. In this project, instead of using Scrum's 30 days iteration, XP's 1 or 2 week iterations were used to get customers' feedback frequently for detail of the process for development of complex projects see Fig. 2 [37].

Combination of the agile techniques is applied at different steps of the projects to achieve flexibility, adoptability. XP, FDD and scrum are together applied to minimize restrictions of the large scale projects [36].

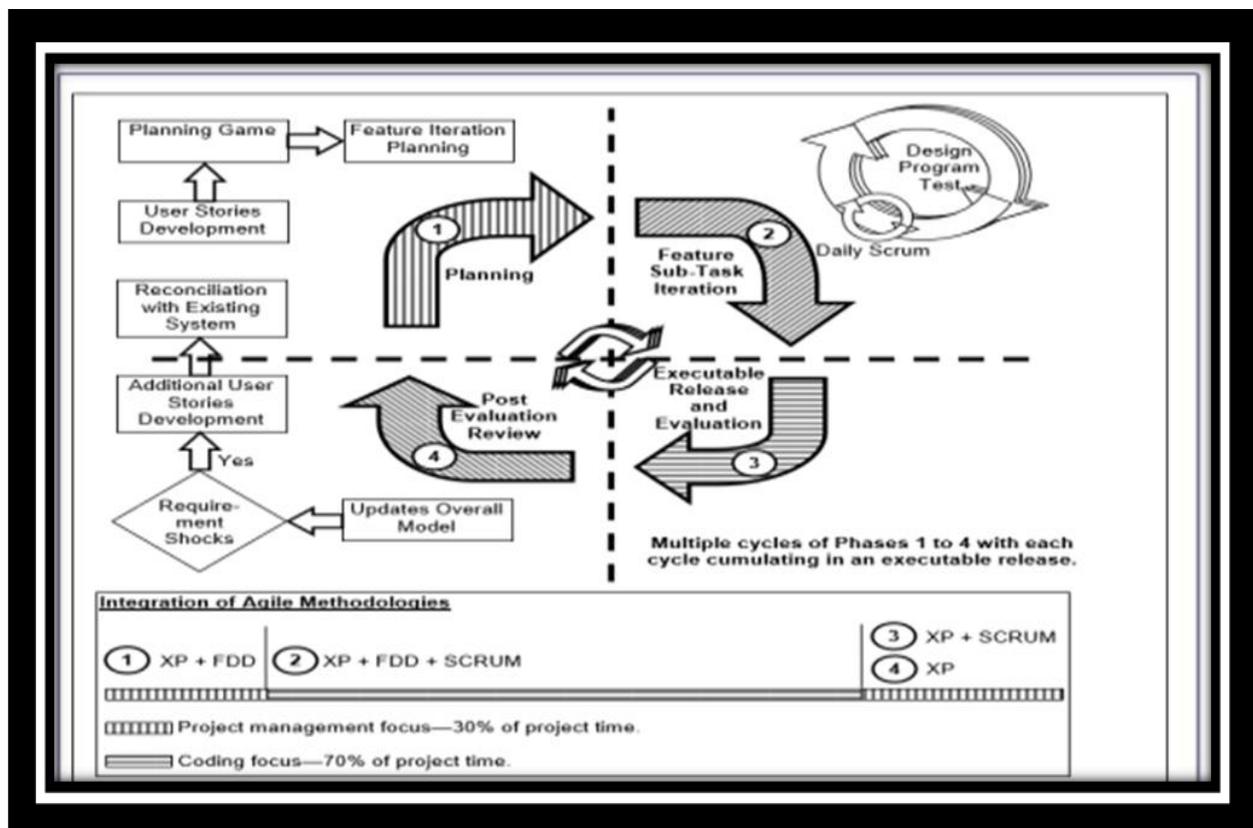


Fig. 3. Framework for agile scalability [38].

For detail of framework for agile scalability see Fig. 3 in this Fig the four phases of the software development is displayed and different agile flavors combination is applied at the all different four phases. Basically the researcher is trying to overcome the each phase limitation while agile is applied by different agile techniques which suit the respective phase.

In the distributed environments and globally distributed teams the communication gap is a big limitation for agile methods adaptations and for covering the problem a framework was suggested that focus on the treatment of the communication problems [39]. Across the borders and limits in globularly distributed teams and in the offshore phenomenon multiple things we have to keep in mind for achieving better results from the projects. All agile methods especially scrum have to be purely agile when applied on the projects so that can be applied on the new domains for their survival [39]. Industry experience reflects the fact that

SCRUM is used to handle the issues like communication, coordination and control in the globally distributed teams'. In return there is evidence that scrum is widely applied in GSD environments. However there are less empirical results found on the usage of the scrum technique in large size projects or in global distributed environments. That's the problem that knowledge is missing between the research and practical implementation of the scrum in GSD [39]. Large size /scale in software industry and projects are popular in IT industry. But it is truth that the heavy weight and larger size /scale projects have more complexities and problems, such as higher risk and unpredictability. Traditional IT project development techniques and management approaches may bring complex IT projects of long time periods, complex inputs and unclear and unwanted results. Thus, it is hard to make customers' satisfaction and recognition easy in the complex IT projects. The agile methods should be applied to complex IT products. Framework for the complex IT projects can be successful for the team coordination and communication [40]. Researchers suggested a hybrid development method for managing larger scale projects that are expected to live for many releases. The hybrid technique employs agile for all new development work. That is the best way to create new purpose in a timely and creative way [41]. Researchers work for agile adoptability in large distributed projects and found that on one side agile is effective for the distributed large size projects while on other hand also facing risk and challenges when applied on the distributed projects, so work is done for the identification of these risks and handling these risks to some extent for details of these risks see Table 1 [42].

Table 1. Risk Categorization Matrix (Dimensions for Segregating the Incidents/Cases Mentioned in Responses) [42].

| S.No | Dimension of Risk Categorization | Description |
|------|----------------------------------|--|
| 1 | Risk factor title | Provisional name given to the risk, some name will be assigned to the risk and the risk will be recognized and represented by that name. |
| 2 | Risk explanation | A brief description of the risk. According to the risk a brief description of the risk will given. In this description the risk will be discussed briefly. |
| 3 | Causes of the risk | Indicate the reason by which the risk is generated. The reason or actors which are generating the risk are stated here briefly. |
| 4 | Source of the risk | Represents the origin of the risk. |
| 5 | Risk management technique | Refers to the method practitioners use to control the risks stated by them. |
| 6 | Literature support | Refers to the set of research papers, which relate to the risk factor identified from interviews. |

In the cited paper researcher presents a case study of a software product company that has successfully integrated practices from software product line engineering and agile software development. We show how practices from the two fields support the company's strategic and tactical ambitions, respectively. It also discusses how the company integrates strategic, tactical and operational processes to optimize collaboration and consequently improve its ability to meet market needs, opportunities and challenges. The findings from this study are relevant to software product companies seeking ways to balance agility and product management. The findings also contribute to research on industrializing software engineering [43]. Agile adaptation is addressed and for this purpose agile adoption and improvement model is designed [44].

5. Conclusion

On the basis of literature review it is found that new techniques of agility is accepted in market for the project's success, and researchers are doing research on agile techniques and are investigating how to extend the benefits of agility towards other domains of the software engineering especially towards large

size projects. While with the changing demand of market it is necessary to adopt the latest methods of agility in large size projects. The research is also providing evidence that agile adaptation also give maturity to the organization standards and helps in the CMM levels achievement. From literature review it is also extracted that very little work is done on the current problem of agile scalability and adoptability in large size projects. All the researchers are talking about the issue but no one is providing some concrete solution to the problem, very few frame works are found to be presented by some researchers.

6. Future Work

Based on identified limitation of agile scalability and current study of existing techniques for agile scalability it is clear that in future there is need to proposed some remedial work in form of model/framework to answer agile scalability problems.

References

- [1] Hamed, A. M. M., & Abushama, H. (2013). Popular agile approaches in software development: Review and analysis. *Proceedings of the 2013 International Conference Computing, Electrical and Electronics Engineering* (pp. 160-166).
- [2] Miller. (2011). The Characteristics of agile software processes, Granville together soft. *Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems*.
- [3] Dot, T., & George, A. W. (2006). Using an agile approach in a large, Traditional project. *Proceedings of the Conference on AGILE*.
- [4] Alistair, C. (2001). *Book Software Engineering*. Editors copyright Alistair Cockburn.
- [5] Dan, T., & Robert, F. (2002). Limitations of agile software processes. *Proceedings of the third International Conference On Extreme Programming and Flexible Processes In Software Engineering (XP)*.
- [6] Ville T. H., Maria, P., & Casper, L., (2013). A mixed-method study of the planning process of a multi-team scrum. *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*.
- [7] Brian, F., Klaas, J. S., Ryan, O., & Donal, O. L., (2013). Scaling agile methods to regulated environments: an industry case study. *The Irish Software Engineering Research Centre, University of Limerick, Ireland* †QUMAS, Cleve Business Park, Monahan Road, Cork, Ireland.
- [8] Moore, E., & Spens, J, (2008). Scaling agile: Finding your agile tribe.
- [9] Maiden, N. (2013). Extending agile processes with creativity techniques. *IEEE Computer Society*, 78-84.
- [10] David, B., George, W., & Des, G., (2013). The maturation of agile software development principles and practice. *Proceedings of the 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems IEEE Computer Society Washington*.
- [11] Aziz, N., Karla, M., Damian, W. I. R., & Hope, A. M., (2013). A case study: Agile development in the community laser-induced incandescence modeling environment. *Proceedings of the 5th International Workshop on Software Engineering for Computational Science and Engineering*.
- [12] Lagerberg, L. S. T., Emanuelsson, P., Sandahl, K., & Stahl, D., (2013). The impact of agile principles and practices on large scale software development projects a multiple-case study of two projects at Ericsson. *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*.
- [13] Ben, O. L., Angin, P., Weffers, H., & Bhargava, B., (2014). Extending the agile development process to develop acceptably secure software, *Dependable and Secure Computing*, 11(6), 497,509
- [14] Antinyan, V., Staron, M., Meding, W., Osterstrom, P., Wikstrom, E., Wrangler, J., Henriksson, A., &

- Hansson, J. (2014). Identifying risky areas of software code in agile/Lean software development: An industrial experience report. *Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE). Proceedings of the IEEE Conference on Software Evolution Week* (pp.154-163).
- [15] Scheerer, A., Hildenbrand, T., & Kude, T. (2014). Coordination in large-scale agile software development: A multiteam systems perspective. *Proceedings of the 2014 47th Hawaii International Conference on System Sciences* (pp. 4780-4788).
- [16] Olsson, H. H., Bosch, J., & Alahyari, H. (2013). Customer-specific teams for agile evolution of large-Scale embedded systems. *Software Engineering and Advanced Applications*.
- [17] Farrow, A., & Greene, S. (2008). Fast and predictable a lightweight release framework promotes agility through rhythm and flow.
- [18] Blau, B., & Hildenbrand, T. (2011). Product line engineering in large-scale lean and agile software product development environments-Towards a hybrid approach to decentral control and managed reuse. *Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security* (pp. 404-408) .
- [19] Subhas C. M., Vinod, K., & Uma, K. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82, 1869-1890
- [20] Minna, P., Outi, S., Raija, K., & Pekka, A. (2011). *Strengths and Barriers Behind the Successful Agile Deployment—Insights From the three Software Intensive Companies in Finland*, Springer Science+Business Media.
- [21] Salo, O., & Abrahamsson, P. (2008). Agile methods in European embedded software development organizations: A survey on the actual use and usefulness of Extreme Programming and Scrum. *Software*, 58-64.
- [22] Soundararajan, S., & Arthur, J. D. (2009). A soft-structured agile framework for larger scale systems development. *Engineering of Computer Based Systems*.
- [23] Murphy, E. H., & Williams, L. (2003). Agile software development: It's about feedback and change, *North Carolina State University, Computer*, 36(6), 39- 43.
- [24] Hajjdiab, H., & Taleb, A. S. (2011). Agile adoption experience: A case study in the U.A.E. *Proceedings of the 2011 IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*.
- [25] Mary, W. (2011). Agile method to improve delivery of large-scale software projects. Faculty of the Department of General Engineering Son Jose State University.
- [26] Ambler, W. (2011). *Methodologist for agile and Lean, IBM Rational IBM agility@scale™: Become as Agile*.
- [27] Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 82(9), 1479-1490.
- [28] Rehan, A., Muhammad, H., & Majid, N. (2008). Agile framework for globally distributed development environment. *Proceedings of the 8th WSEAS International Conference on Applied Informatics and Communications*.
- [29] Begel, A., & Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. *Proceedings of the First International Symposium Empirical Software Engineering and Measurement* (pp. 255-264).
- [30] Jianbin, S. T., & Fang, D. (2009). A heavy weight IT project management framework based on agile theory. *Proceedings of the International Conference on Management and Service Science*.
- [31] Supriya, V. S., & Urvashi, R. (2015). Categorization of risk factors for distributed agile projects. *Symbiosis Centre for Information Technology (SCIT)*.
- [32] Frank, K. Y. C., & James, Y. L. T. (2009). Acceptance of agile methodologies: A critical review and

conceptual framework. Department of Information Systems, Business Statistics and Operations Management, School of Business and Management, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong .

- [33] Kuda, N. R., Kavita, G. N., & Praneeth, C. (2011). A study of the agile software development methods, Applicability and implications in industry. *International Journal of Software Engineering and its Applications*.
- [34] Shen, M. J., Yang, W. R., Rong, G. P., & Dong, S. (2012). Applying agile methods to embedded software development: A systematic review. *Software Engineering for Embedded Systems (SEES)*, 30-36.
- [35] Smits, H, (2007). The impact of scaling on planning activities in an agile software development center. *Proceedings of the Conference on 40th Annual Hawaii International*.
- [36] Gabrielle, B. (2008). Rolling out agile in a large enterprise. *Proceedings of the 41st Hawaii International Conference on System Sciences*.
- [37] Deepti, M., & Alok, M. (2011). Complex software project development: Agile methods adoption. *Journal of Software Maintenance and Evolution: Research and Practice*.
- [38] Chuan, H. T., Wee, K. T., Hock, H. T., Hock, & H. T. (2013). *Designing An Information Systems Development Course to Incorporate Agility, Flexibility, and Adaptability* .
- [39] Abbass, G. (2010). Success of agile environment in complex projects. *Proceedings of the Conference on Edith Cowan University Research Online Australian Information Warfare and Security*.
- [40] Lehman, T. J., & Sharma, A. (2011) , *Software Development as a Service: Agile Experiences*.
- [41] Read, A., & Briggs, R. O. (2012). The many lives of an agile story: Design processes, design products, and understandings in a large-scale agile development project, system science. *Proceedings of the 2012 45th Hawaii International Conference on System Sciences*.
- [42] Geir, K. H., & Tor, E. F. (2007). Process fusion: An industrial case study on agile software product line engineering. Department of Computer and Information Science.
- [43] Qumer, A., & Henderson, S. B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. Faculty of Information Technology, University of Technology, Sydney, Australia.
- [44] Xiao, F. W., & Conboyb, K. (2012). Leagile” software development: An experience report analysis of the applicationof lean approaches in agile software development. *The Irish Software Engineering Research Centre, Ireland The Journal of Systems and Software*.



Hina Saeeda was born in 1989. She is currently doing MS degree in software engineering from Miltry College of Signals NUST Pakistan. She received BS degree in 2011 from UIIT-PMAS-AAUR pakistan. Her research interest is software engineering.



Fahim Arif was born in 1967. He is currently professor at MCS-NUST Pakistan. He received his master degree in sciences computer software engineering from National University Science and Technology, Islamabad in 2003. He has completed his PhD degree from National University Science and Technology in 2009. His research interests are quality assurances, software engineering and image processing



Nasir Mehmood Minhas was born in 1973. He is currently an assistant professor in UIIT-PMAS-AAUR. He is currently pursuing his Phd degree from MAJU Islamabad Pakistan. His research interest is software Engineering.



Mamoona Humayun was born in 1983. She is currently an assistant professor in UIIT-PMAS-AAUR. She did PhD from Harbin Institute of Technology Harbin in China. She is currently pursuing his PhD degree from COMSAT University Islamabad Pakistan. Her research interests include global software, development 12requirement engineering.