Architecting Activities Evolution and Emergence in Agile Software Development: An Empirical Investigation

Initial Research Proposal

Muhammad Waseem^(✉) and Naveed Ikram

Riphah International University, Islamabad 44000, Pakistan
m.waseem@iiu.edu.pk, naveed.ikram@riphah.edu.pk

Abstract. This proposal is design to address the proposed research work on agile software development and architecture co-existence. The objective of this research is to answer how architecting activities emerge and evolve with agile software development in industry. The architecting activities are architectural analysis (AA), architectural synthesis (AS), architectural evaluation (AE), architectural implementation (AI), architectural maintenance and evolution (AME), architectural recovery (AR), architectural description (ADp), architectural understanding (AU), architectural impact analysis (AIA), architectural reuse (ARu) and architectural refactoring (ARf). This research objective could achieve by using multiple research methods. We are planning to use comprehensively report the pure ‘state- of- practice’ for architecting activities in ASD from industry and practitioners point of views. Therefore, we decided to use the case studies, survey and semi structure interview as research methods. The result of this research work can provide the baseline information for architecture evolution frameworks for agile software development, challenges and solutions in ASD for SA activities, expected evolvable dimensions of the software system, methods that may help for minimizing the architectural and agile co-existence issues and architectural technical debt in agile software development.

Keywords: Software architecture · Agile development · Architecting approach

1 Introduction

Agile methods widely accepted by the software organization in reaction of heavyweight software development processes. Agile software development(ASD) respond to the changes, people collaboration and working software instead of emphasizing on bureaucratic and upfront planning [1]. Many classical software development activities can align with agile software development such as requirements, architecting, coding, testing and deployment. However, researcher have doubt that practitioners do not pay the sufficient attention to architectural activities in agile software development [2, 3]. Software architecture and agile related research reports two extreme views [4]: First, upfront design and SA evaluation are highly time and effort consuming activities therefore you don’t need to go with architecture centric activities, refactoring would help to resolve most of

structural problems [2, 5]. Second, the proponent of SA is sure that the sound architectural practices (SA analysis, design, description or documentation) cannot be followed completely in agile software development, which may effects on the project quality. Naturally, *a question arises then what happen with architecture in agile software development?*

In recent years, a good number of studies have investigated the architectural related challenges and solutions in agile software development [5–9]. These studies focus on different perspective of software architecture for example a study of Boehm identify the organizational and technical challenges that involved in integrating the traditional and agile software development process [8, 10]. Acuna et al. reports agile methods do not pay sufficient attention to the architecture centric activities as compares to traditional process so that's we could not found the significant guidance on SA activities in agile context [11]. MA Baber identifies the architecture related challenges and issues which agile teams could face. It has been observe that, architects should have the sound skills and knowledge about implementation domain [12]. Boehm argues for hybrid approach for agile architecting and development. According to him, combine the necessary characteristics from agile and plan driven development for projects implementation.

The aim of this research is to empirically investigate: how software architecture and agile used in combination and how architecture emerge and evolve in agile software development.

2 Problem Statement and Motivation

Software evolution may analyze through different ways; for example releases histories, source code analysis and architecture analysis. This proposes research plan focus on architecting activities evolution and emergence in agile software development. First, software architecture provides the base to software system [13]. Second, the architecture of software system presents the high level structure and behaviors of the system which are expected to evolve with passage of time [14] and provide bases for evolution [15]. Thirdly, it is supposed to be agile and software architecture is proponent to each other. For example, if teams spend too much time on software architecture it may possible working software may delay. If teams pay little time to SA then the team may face high risk of system failure (how much upfront). Fourth, SA and agile combination received the significant consideration in recent years for research but there are very less number of studies that consider the architecting activities emerge and evolve [16]. Therefore, we decided to come up with following problem statement.

Problem statement: “How does architecting activities emerge and evolve in agile software development?”

Software architecture consists of numbers of activities that explain the process and stages of software architecture. We have select following activities from systematic mapping study [16] and formulated in a questions for our proposed research.

- RQ 1: How does an architectural significant requirement are identified and maintain in agile software development?*
- RQ 2: How does architectural solutions are be provided for ASRs in agile software development?*
- RQ 3: How does scenario base architectural evaluation (AE) is conducted in agile software development?*
- RQ 4: How much detailed design is enough in agile software development?*
- RQ5: How does architecture emerge and evolve in agile software development?*
- RQ6: How does architecture maintain in agile software development?*
- RQ7: What are the possible benefits of architectural recovery (AR) in agile software development in term of quality time and cost?*
- RQ8: What are most useful architectural views are being used in agile software development and why?*
- RQ9: How does Architectural Understanding (AU) is used to comprehend the architectural elements (e.g. architectural decisions) and their relationships in an architecture design for agile software development?*
- RQ10: Does Architectural Impact Analysis (AIA) really have worth in agile software development?*
- RQ11: How does existing architectural reusable components such as architecture frameworks, decisions, and patterns are used in agile software development*
- RQ12: How does Architectural Refactoring (ARf) is happen in agile software development?*

3 Description of Proposed Research Methodologies

Architecture is very much depended on architect and teams so we need to investigate the people and there interactions over the process (architecting in agile). We are covering major architecting activities in agile software development that may identify by using different research method. So we have decided to use qualitative and quantitative research (where required) method for evaluating our research questions. We will design the case studies, survey and semi structure interview for exploring the practitioners experience about architecture evolution in agile software development. We may also use the experiments for particular architecting activities such as in architecture evaluation.

3.1 Survey

Considering the objectives of our research and available resources, we could go with survey research method to understand the architectural and agile practices from architecting activities perspective. A survey research method is considered suitable for gathering self-reported quantitative and qualitative data from a large number of respondents [17]. Our survey design will be a cross-sectional. Survey research can use one or a combination of several data gathering techniques such as interviews, self-administered questionnaires and others [18]. Our possible method will be

questionnaire as a data collection instrument because we want to obtain the information from a relatively large number of practitioners, many of whom we would not be able to contact personally. Our proposed survey may consist on following activities.

- Instrument construction and evaluation
- Instrument deployment
- Target population identification
- Instrument deployment
- Sampling techniques selection and invitation mechanism
- Data validation and data analysis

3.2 Case Studies

Our proposed research questions may answer through ‘Multiple Embedded Case studies’ from industry and practitioner. Following are the generic outline that may follow for achieving our research goal(s).

- Devising unit of analysis
- Deciding case selection criteria
- Data collection technique and process
- Setting up population
- External validity
- Reporting study limitation
- Scheduling and
- Reporting

We are interested to apply both Primary and secondary data collection techniques on collected data. Our data may consist of on field notes, audio recordings of meetings and discussions, photographs and copies of artifacts. We will apply the triangulation approach to incorporate multiple vantage points. We can achieve this diversity by using different data sources and types, and by engaging multiple observers. Additionally, we are interested to discuss our findings with the respective teams for initial verification.

3.3 Semi Structure Interviews

Architecture is very much depended on architect and teams so we need to investigate the people and there interactions over the process (architecting in agile). Semi structure interview is good technique when depth is required for particular phenomena. Interview question will be design before taking the interview from participant(s). It would not be not necessarily to ask questions in same order as they are listed. So we would ask the questions according to situation. Further, semi structure interview allow ‘improvisation and exploration’ in study subject. During the interview session, we will record the participant response in audio/video format and we will take the notes where things need to write. Our focus would be, how individuals/team qualitatively and quantitatively experience about architecting activities.

Our potential participant would be experienced architect, senior developer, team leads and those who have significant development experiences in agile way.

4 Data Analysis Method

Data analysis methods are different for qualitative and quantitative data. Our collected data may consist of large amount of qualitative data, so we are decided to analyze this data through constant comparison method that originally presented by Glaser and Strauss [19], it has been practically explained by the some other [20]. We will use the guide lines that has been presented in [20] for constant comparison method. Steps involved for our data analysis are

- Preformed coding field notes periodically
- Grouping into patterns according to code
- Writing of field notes

For quantitative data, analysis may include descriptive statistic (mean values, standard deviations, histograms, scatter plots etc.), correlation analysis, development of predictive models, and hypothesis testing [21].

5 Future Agenda

This research study could provide information on the issues of agile architecture co-existence including how architectural analysis and description change over the time? Further, this study would also be exploring architectural models evolution on different level, particularly on system level. This could be beneficial for analyzing traceability between changing requirements, features and architectural model to improve the evolution process. Furthermore, this study would also be analyzed the architectural and design pattern evolution, this would expectedly heighten the awareness about different kind of design and architectural practices and there possible threats in agile software development. To the future researchers, The result of this research can provide the baseline information for architecting activities frameworks for agile software development, expected evolvable dimensions of the software system, methods that may help for minimizing the architectural and agile co-existence issues and architectural technical debt in agile software development.

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. Kent Beck, M.B., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development, Feburvery 2001. <http://agilemanifesto.org/>
2. Abrahamsson, P., et al.: Mobile-D: an agile approach for mobile application development. In: Companion to the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages, and Applications. ACM (2004)
3. Boehm, B.: Get ready for agile methods, with care. *Computer* **35**(1), 64–69 (2002)
4. Babar, M.: Making software architecture and agile approaches work together: foundations and approaches (2014)
5. Nord, R.L., Tomayko, J.E.: Software architecture-centric methods and agile development. *IEEE Softw.* **23**(2), 47–53 (2006)
6. Abrahamsson, P., et al.: Mobile-D for Mobile Software: How to Use Agile Approaches for the Efficient Development of Mobile Applications. VTT Technical Research Centre of Finland, Finland (2005)
7. Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*, 528 p. Addison-Wesley Longman Publishing Co., Inc., Boston (2003)
8. Lycett, M., et al.: Migrating agile methods to standardized development practice (2003)
9. Parsons, R.: Architecture and agile methodologies-how to get along. In: Working IEEE/IFIP Conference on Software Architecture. IEEE Computer Society, Vancouver (2008)
10. Boehm, B., Turner, R.: Management challenges to implementing agile processes in traditional development organizations. *IEEE Softw.* **22**(5), 30–39 (2005)
11. Acuña, S.T., et al.: A systematic mapping study on the open source software development process. In: 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012). IET (2012)
12. Babar, M.A.: An exploratory study of architectural practices and challenges in using agile software development approaches. In: Joint Working IEEE/IFIP Conference on Software Architecture, 2009 & European Conference on Software Architecture, WICSA/ECSA 2009. IEEE (2009)
13. Clements, P., Kazman, R., Klein, M.: *Evaluating software architectures* (2003). 清华大学出版社
14. Garlan, D.: Software architecture: a roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*. ACM (2000)
15. Medvidovic, N., Taylor, R.N., Rosenblum, D.S.: An architecture-based approach to software evolution. In: *Proceedings of the International Workshop on the Principles of Software Evolution* (1998)
16. Yang, C., Liang, P., Avgeriou, P.: A systematic mapping study on the combination of software architecture and agile development. *J. Syst. Softw.* **111**, 157–184 (2016)
17. Kitchenham, B.A., et al.: Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Softw. Eng.* **28**(8), 721–734 (2002)
18. Lethbridge, T.C., Sim, S.E., Singer, J.: Studying software engineers: Data collection techniques for software field studies. *Empirical Softw. Eng.* **10**(3), 311–341 (2005)
19. Glaser, B.G., Strauss, A.L.: *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Transaction Publishers, Piscataway (2009)

20. Miles, M.B., Huberman, A.M.: *Qualitative Data Analysis: An Expanded Sourcebook*. Sage, Thousand Oaks (1994)
21. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.* **14**(2), 131–164 (2009)