

A REVIEW ON SOFTWARE ARCHITECTURE DOCUMENTATION IN AGILE DEVELOPMENT

Awais Rasool¹, Manzar Hussain²
University of Agriculture, Faisalabad, Pakistan
awaisuaf11@gmail.com, manzerhusain375@gmail.com

ABSTRACT— With the growing development of world every software product demands extra features to compete with competitors. To solve this problem, we have required Agile with defined and clear architecture to avoid the failure of project. But you do not know how it is possible and how we can stable our architecture and Agile model...? To ease the solution of problem we divided our product development models into two sections. 1. Traditional Models and 2. Agile Models
The traditionally models are highly time consuming with dense documentation as compared to agile with frivolous documentation and intercommunications to satisfy customer. There are different methodologies available for starting products and complex systems development consisting the simple requirement with well-structured architecture objects. But we choose the agile process to make our architecture slim and least documentation which can easily review, update and conversate. There are also problems in our documentation and architecture design but we propose solutions with problems for the easiness of reader. We also propose different solutions in documentation problems.

Keywords— Agile; Software Architecture; Quality Attributes; Software Design; Documentation; Software Quality.

I. INTRODUCTION

Software architecture is qualitative to improve the quality of system, further reusing of software Architecture knowledge in cross project software, maintenance of software and development. In previous decade mostly traditional models were used include (waterfall, Prototyping, RAD, Spiral Model, Unified Process and incremental). While we use the Agile models includes but they results in late completion of project due to dense documentation.

Solution of this problem is to use software architecture skills in agile method for the production of satiable quality regarding a software system to make the architecture specification documentation slim and least which facilitates in evaluation, updating and communication the documentation to avoid from further interruptions like traditional models. Hence architecture play important role in documentation.

Agile models are highly acceptable and become popular in industry and work iteratively which facilitates in changing requirements with solid and regular conversation between stakeholders and software developers during the software developing process.

It has low risks, few management cost develop excellent team environment and strongly focused on project objectives with high customer engagement. Thus agile development provides the source of communication between stakeholders and machine. We purpose the problems with documentation and architectural design in agile software development and their solution.

II. ARCHITECTURE DOCUMENTATION

Architecture of any project is a key part of documents of software development process.so many different types of documents are used to support to maintain consistency. When we are creating Architectures and different types of design choices available and different facilities are also help to contain this type of documentation.

Architectural documentation is also a key process to success of any development process. The main purpose of is to provide training facilities to make well the intended system.in this process motivation is also play key role in documentations. Architecture document is satisfiable and tangible to provide services as a blueprint so for creating as an information. Recently literature on architecture documentations that communication to heavily weight development process model. In this type

of methodologies the initial iterative development model underlined the construction of a hard steady routine timely products life cycles. This looks to be risky factor in the achievement of Structure huge long life software system.

What type of questions generate our mind architecture a vast design? There are several agile procedures to choose from and qualified on each one Dreadful on the features of architectures. This segment discovers opportunities for educational design and documentation problems in agile progress. We study the accuracy of standard methods and recommended more than results that might price further research. Our specialized learning as greatly energetic and 50% of then Deliberate that new & agile techniques of educational documentations should be investigation process more sensibly. We have to use research methods to improve our Documentation process in agile methodology. There are couples of traditions to improve our Architectural documentations. We can say that every documentation of architecture play well role. We would like to manner a more detailed analysis of class complications.

RELATED WORK:

There are sections of this part of paper below.

A. software architecture and agile development:

From many years, a lot of confusion in software architecture role and their importance in agile approaches. To resolve this problem someone give solution that every software system architecture has intensive in nature, regardless the process to develop the system and they define that how much architecture is needed for a given development project using eight dimensions like semantics, scope life cycle and cost etc. This all done by workshop, where the different opposing group share their ideas to give roadmap. Moreover, these all workshop helps to define a common model of various practices that contribute to the project success. In all studies our main focus on software architecture design activity in agile method to analyze quality attributes of software

systems.

B. Combine software architecture and agile development:

This section is quality evaluation method for an agile approach based on quality attributes to helps participants to identify quality issues in the early phase of a project and obtain feedbacks for quality improvement.

This approach consists of 5 phases:

First phase: The critical quality attributes of a system

Second phase: rank quality attributes

Fist second and third phase are executed in the early stages of a process. When the development cycle gets started, quality requirements are elicited from related functional requirements

Fourth Phase: Assess the quality attributes by quality requirements

Fifth phase: Feedbacks are obtained from evaluation results

That type approach helps participants to focus on attribute quality that able to make plan on quality improvement. To have more trust in the result participants have creating quantitative evaluation model

METHODOLOGIES

There are two background of our study methodologies in this section.

A. Interviews about Agile:

We use well designed qualitative research interviews method consisting of same types of questions about agile development in agile methodology which enhance the future research demands, pros and cons, benefits and qualitative features in architectural design and documentation. There were two forms of questions in interview (printed and electronic) with text free and multiple choice answers. Duration of interview was 10 minutes which has three parts.

1) Questions about experience, profession and field of profession.

2) General Questions

3) Establish the survey list

B. Interviewees' Background:

There were 40 total participants in survey. Ten

participants were consider themselves not experts. Three interviews were not completed their interviews.31 were professional experts and 6 were industrial experts. The industrial experts shown their experience because they were group leaders and senior developers in industries and we found academic background from PhD students to full professors. Every participants in interview shown his interest about agile activities like management and theoretical base in whole SDLC (software development life cycle).The expert's participant (65%) ensured the use of agile in past and were planning to usage them in future and around 20% were planning to use them in future.

At last 12% participants have not any interest to use agile in future and only one used agile in past but would not use it again. From our survey we concluded that a lot of participants had a positive approach and practice through development .Agile is mostly used in Scrum (49%), Open source development (32%) and Extreme programming (22%) while the open source is not purely agile based as Rule of SE but it has use it in design and customer contribution activities during development in our documentation emerging us toward our goal but if this not happen then we should follow Agile Manifesto rule about agile method. Agile is a main pillar for working of a project and it helps in architecture documentation, specification, and approval processes.

C. How to Fit Architecture in Agile:

In Agile architecture should work in iterations and provide well guided directions to team which leads our project to achieve goal and provide them relaxation while developing the project. The iterative approach keep synchronization in design, give starting point of project and collecting feedback from stakeholders. Agile compromise of sprint review meetings that clear the better picture of entire application and retrospectives are held to estimate what is being built and is it working well. Caring factor during documentation:

- Assimilate project requirements with creative requirements and standards which give good base of project
- For avoidance unreliability we should use quality tools and security tools

- Conduct daily basis collaborative meeting with teams to assure that our architecture is not only paper work but it actually fulfilling reality base needs of stakeholders
- Include all team members in our architecture development to meet their needs of understanding about project

PROBLEMS IN AGILE DEVELOPMENT:

The following portion have describe different problems of agile development based on interviews.

A. Problems and the Essence of Agile Development

This is primary portion of all we personally requested from expert that how personally they look the main disadvantages of agile development. However mostly expert suggested that agile development method applied on that projects that have certain characteristics. These problems cannot improved through process modifications to agile development but require new approach. Moreover 13.5% of the experts, may have 20 years of experience in their academia, agreed the lack of formalities in documentation and architectural design.

B. A Detailed Investigation of the Problems

This portion focus on source code problems which is detected by experts in SDLC because they were professional in design and documentation because source code is the source of communication between developers which is more important that is why they are sensitive to code problems. Hence we concluded that with more work experience participants faces minor problems with documentation they saw and improved architecture design probably (34%).

THE ORIGINS OF THE PROBLEMS:

The Origin of the problem heading explain detailed overview of Architecture documentations. In the section we can define the origin of the difficulties in expert opinions. Different types of issues terminologies in different stages your method is always strange to agreement this

category of impediment. This Part exposes that the typical of specialists recognizes a good fortune of accreditations and architectural design in agile development. Therefore we requested the some professional about favorable causes in design certifications and code. Generous to our professional, the causes for potential issues for humble design are that designer impact not care about 49%. Insignificant reasons for humble documentations and ignored architectural design are that some other conflicts with the goal of script well-meaning documentations. For senior professional, the key problems is that certification and policy are not clearly considered. In overall, more knowledge able meetings approve fewer interpretations for the being of difficulties and nobody of the utmost specialists declared nonexistence of time as a goal. Professional attending in early SDLC and progress consider unacceptable time, luck of awareness' not attentive about and imperfect personal paybacks as the main reasons for unfortunate certifications and plan. Authorities attentive in organization deliberate that lack of knowledge.

The Proposed Solution:

Our documentation contains two solutions regarding agile documentation

A: Abstract Architecture Specification

Document B: Abstract Specification Tool

Abstract Architecture Specification

Document:

The abstract architecture specification (AAs) document consisting of related and efficient information of architecture as well as having the status of products in architecture

There are four parts of our solution which are given below:

- Product overview
- Product goals for upcoming release
- Product architecture overview
- Non-functional requirements

Conceptual layers in our architecture documentation catering for different stakeholders.

- System components layer
- Common and cross concern components layer

- External integration components layer
- Functional components layer

In functional component layer picture of functional architecture layer is included in document and components are presented in text form, a list of acronyms used to links relevant documents. We suggested template based architects to make our documents shorter and attach related data in our defined field instead of structured free text.

The AAS document provide the facility of communication between architect and stakeholder due to easy language. Any reviewer can easily concentrate on content and design decisions instead of long described statements. Due to this property of our document we can get our desired comments from reviewers by following the changes in architecture with passage of releases and can check development of architecture.

Abstract Specification Tool:

The main benefit of abstract specification tool is to decrease the stress of work in documentation by arranging well-structured and related information in documents for creation of design and a clear picture of architecture. It facilitates in two factors (form to fill and modeling tool) which allows abstract specification tool to upgrade as essential through forms or directly by modeling tool. This decreases the efforts to manually insert, reuse information's and designs throughout the design phase. It also allow architects to organize their works using checklists which is combined within tool and lead the architects throughout the development and results in generations of architecture documentation over time.

Future Work:

In future we are scheduling survey with grater group of participants to enhance the participation of industry experts. In our recent survey we include only 37 industry experts to conduct their views about agile development which not satisfy us.

CONCLUSION:

In this paper we discuss the architecture documentation using agile method and recognize the standard and well organized methods in

development of software to ensure the stakeholder satisfaction, volatile requirements and deliver software in minimum time. Collect the expert's views about agile development and provide a detailed analysis and root causes of agile. The problems discussed in our documentation are not specific to agile but they are common to all development strategies. We proposed many solutions about agile problems and concentrate on design of software to improve the quality attributes (that are helpful for projects) of all types of software. We introduced new methods not use the traditional methods of documentation to combine software architecture with agile methods.

References:

- [1] R. Barbacci, R. Ellison, A.J. Lattanze, J.A. Stafford, C.B. Weinstock and W.G. Wood, Quality Attribute Workshops (QAWs), Technical Report, 3rd Edition, CMU/SEI-2003-TR-016, Software Engineering Institute, Carnegie Mellon University, 2003.
- [2] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord and B. Wood, Attribute Driven Design (ADD), Technical Report, Version 2.0, CMU/SEI-2006-TR-023, Software Engineering Institute, Carnegie Mellon University, 2006.
- [3] R. Nord, M. Barbacci, P. Clements, R. Kazman, L. O'Brien and J. Tomayko, Integrating the Architecture Tradeoff Analysis Method(ATAM) with the Cost Benefit Analysis Method (CBAM), Technical Note, CMU/SEI-2003-TN-038, Software Engineering Institute, Carnegie Mellon University, 2003.
- [4] Taehoon Um, Neunghoe Kim, Donghyun Lee, Hoh Peter, A Quality Attributes Evaluation Method for an Agile Approach, ACIS/JNU-2011- International Conference on Computers, Networks, Systems, and Industrial Engineering
- [5] G. Booch, "An Architectural Oxymoron," IEEE Software, vol.27, no. 5, pp. 95-96, 2010.

- [6] L. Bass, P. Clements and R. Kazman, "Software Architecture in Practice". 3rd ed., Addison-Wesley Professional, 2013.
- [7] I. Hadar and S. Sherman "Agile vs. Plan-Driven Perceptions of Software Architecture", 5th Int. Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), ICSE, June 2012.
- [8] J. Tyree and A. Akerman, "Architecture Decisions: Demystifying Architecture". IEEE Software, vol. 22, no. 2, pp. 19-27, 2005.
- [9] R. Barbacci, R. Ellison, A.J. Lattanze, J.A. Stafford, C.B. Weinstock and W.G. Wood, Quality Attribute Workshops (QAWs), Technical Report, 3rd Edition, CMU/SEI-2003-TR-016, Software Engineering Institute, Carnegie Mellon University, 2003.
- [10] R. Wojcik, F. Bachmann, L. Bass, P. Clements, P. Merson, R. Nord and B. Wood, Attribute-Driven Design (ADD), Technical Report, Version 2.0, CMU/SEI-2006-TR-023, Software Engineering Institute, Carnegie Mellon University, 2006.
- [11] R. Nord, M. Barbacci, P. Clements, R. Kazman, L. O'Brien and J. Tomayko, Integrating the Architecture Tradeoff Analysis Method(ATAM) with the Cost Benefit Analysis Method (CBAM), Technical Note, CMU/SEI-2003-TN-038, Software Engineering Institute, Carnegie Mellon University, 2003.
- [12] Taehoon Um, Neunghoe Kim, Donghyun Lee, Hoh Peter, A Quality Attributes Evaluation Method for an Agile Approach, ACIS/JNU-2011- International Conference on Computers, Networks, Systems, and Industrial Engineering
- [13] M. A. Babar, "An exploratory study of architectural practices and challenges in using agile software development approaches," in WICSA/ECSA 2009., 2009, pp. 81 – 90.
- [14] O. Salo and P. Abrahamsson, "Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of xp and scrum," Software, IET, vol. 2, pp. 58 – 64, 2008.
- [15] B. Sheth, "Scrum 911! using scrum to overhaul a support organization," in Agile Conference, 2009, pp. 74 – 78.