Usage of Software Design and Architecture in Agile Teams

Bijayalaxmi Panda
Arizona State University
bpanda1@asu.edu

Divya yadamreddi
Arizona State University
dyadamre@asu.edu

Abstract

Agile, now a day is the most trending and adaptable standard in the software industry. Agile development methods are mostly used as a software delivery process and have been widely used since its inception. Agile aims to deliver the software product incrementally from the starting of the project instead of delivering the whole product at once at the end. Teams following the agile methodology collaborate with cross-functional disciplines to deliver. They work within a common framework that guides for the end product. Each step of software development like planning, integration, deployment and release of the product are done by the teams together. To work smoothly a simple, flexible, and adaptable software design and architecture is necessary.

Keywords Agile, Architecture, Requirements, Design, Design Principles, Incrementality, Sprint

ACM Reference Format:
Bijayalaxmi Panda and Divya yadamreddi. 2019. Usage of Software Design and Architecture in Agile Teams. In *SER574 '19: Advanced Software Design, January 20, 2019, Mesa, AZ*. ACM, New York, NY, USA, 2 pages. https://doi.org/12.1337/1122456.1122456

1 Introduction

An agile team is a cross-functional group of people that have everything and everyone, necessary to produce a working, tested increment of product. Agile encourages a disciplined project management process that encourages frequent inspection and adaptation. A leadership philosophy that supports teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.[4]

Software architecture is the design and specification of the rules by which software will be built. It could be as high level as possible to give the overview of the project to extend the product. If the software has a lot of isolated components, then the architecture ensures that the system operates in an environment that supports components with isolated dependencies. It is more about the collective understanding of how a system should be built to respond to the change.

Software Design act as a blueprint during the development. Having a correct and updated software design is must for the teams to work in an agile method. Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as in a stylized software engineering process.[1]

In this paper, Section 2.1 describes the general idea of agile development. Section 2.2 focus on the primary steps involved in software design and architecture. The section 2.3 focuses on the impact of software design and architecture teams working in agile.

2 MAIN

2.1 General Idea of Agile Development

Agile methodologies are a group of software development methods that are based on iterative and incremental development. The four major characteristics that are fundamental to all agile methodologies are adaptive planning, iterative and evolutionary development, rapid and flexible response to change and promote communication. Agile consists of several iterations which include- 1. Planning 2. Requirement Analysis 3. Design 4. Coding 5. Unit testing 6. Acceptance Testing[5]

2.2 Software Design and Architecture

Architecture in agile can be defined as a specific set of rules by which software will be built. It can also be defined as the design by which the components of a system will behave and interact. Architecture can be described in high level or detailed. High-level meaning, it could just specify as the software is going to be built using APIs or detailed meaning giving the details of the technology stack of each and every component.

There are a number of reasons to explain why architecture is important in Agile methodologies. A software project involves a high cost and to mitigate the chances of increasing the cost, it is important for the software project to have a proper architectural design in its early stages. A specified architecture would not only give the form to a project but also acts as a preliminary test to see if the requirements of the customer have been properly incorporated by the development team. The architecture is also important in a way that without it, the project cost, schedule and quality cannot be predicted correctly. It also determines the key approach to the choice of tools, work-breakdown, skills and technologies that are required to finish the project.

Software Design is both a process and model. One of the main components of software design is the software requirement analysis. Depending on the software, software design can be as simple as a flow chart or defined in a model like Unified Modeling Language (UML).

The architecture and design of a project are important in an agile environment. Hence it should be created in such a way as to accommodate the changes in the development phases such as as[3]

- The requirements in any project are not stagnant and keep changing and evolving with respect to users' needs, advancement in technology and also some constraints with respect to time, budget and labour.
- These requirements change automatically leads to changes in software design and development too (code).
- Due to the high budget of the project, accommodating all these changes would, in turn, lead to a budget increase or cutting the cost of labour. This could be considered an expensive critical activity in rapidly changing requirements.
- The architecture should also specify a clear software design process so as to avoid any miscommunication in terms of development which in turn might lead to bigger changes incurring higher costs.

2.3 Impact of Design and Architecture on Agile Teams

In agile, the architecture of software is developed in an iterative manner, beginning with enough definition for the team to start design and development. The architecture guides the team in a direction that gives the best chance for success.

There is a trade-off between allowing a team to choose what makes sense for their project while ensuring they don't pick technologies that are incompatible with the rest of the enterprise. This trade-off makes sure the team builds software that aligns to the overall enterprise architecture.

Like any mature agile process, an agile approach to architecture relies on doing just enough definition up front to get started, gathering feedback as we go, adjusting as needed, and iterating frequently to keep architecture and design in sync with the emerging application.

Prior to the first sprint, a high-level enterprise or system architecture should be created (if it doesn't already exist) and discussed. As part of each sprint kickoff, a team-based design is used to update the as-is design from the last sprint to account for feature additions and enhancements that will be made in the current sprint. As stories are tasked out for development and testing, the team will use this new "to be" design to help determine what will need to be implemented in order to satisfy the expected design.

Sprint review meetings should present how each story fits into the bigger picture of the whole application and the enterprise. Sprint retrospectives should include architecture and design in the discussions of what is working and what isn't.

In all facets, architecture should be treated like any other part of the agile process:[2]

- By starting with experienced architects and industry best practices, we can integrate the project requirements with the enterprise requirements and standards to be confident in a good foundation
- By incorporating retrospection and review in the process, we can make minor adjustments to both the architecture and the software being developed to ensure that we meet the enterprise's needs
- By driving that feedback using quality tools, security tools, testing, and other objective metrics, we ensure that we aren't reacting to guesses and anecdotal evidence
- By reflecting and evaluating continuously, the course corrections remain minor and incremental and can adjust to changing architecture requirements throughout the life of the project and as needs evolve
- By collaborating with the development teams, we can be sure the architecture reflects real-world needs and isn't just an "ivory tower" architectural approach that makes sense on paper but fails in actual use.
- For an organization transitioning to agile development, creating software architecture isn't incompatible with your new processes. Consider the principles in the Agile Manifesto, involve team members who will be using the architecture in its development, and reflect and adapt often, and you will end up with an architecture that meets the needs of your team and your enterprise.

3 Conclusion

Software development projects which involve high speed, rapid changes and huge budgets not only need a good agile practice or methodology to finish the project successfully but also needs good architecture and design practices to finish the project on time and at expected costs. Architecture and design of a project go hand in hand and are important so as to give the project the sense of direction that it needs.

References

[1] Peter Freeman and David Hart. 2004. A Science of Design for Software-intensive Systems. *Commun. ACM* 47, 8 (Aug. 2004), 19–21. https://doi.org/10.1145/1012037.1012054

[2] Gene Gotimer. 2017. An Agile Approach to Software Architecture. https://www.agileconnection.com/article/agile-approach-software-architecture

[3] Sandeep Chinthapatla Mounika G Mehdi Mekni, Gayathri Buddhavarapu. 2018. Software Architectural Design in Agile Environments.

[4] Carlos Pliego. 2018. Software Architecture—In an agile environment. Retrieved February 03, 2018 from https://medium.com/@carlospliego/agile-software-development-and-architecture-3dbb243fcbd2

[5] Divya Yadamreddi. 2018. Agile Practices.