Enabling Agile Through Teamwork

Cecilia La Place*
Chiranjeevi Ramamurthy*
claplace@asu.edu
Arizona State University
Mesa, Arizona

Abstract

For more than a decade, the Agile development process has seeped into the lives of software developers and customers, changing the way projects are planned, how teammembers and teams interact, and how customers receive their product. Agile is a teamwork heavy process, demanding superb communication and technical skills to develop projects where requirements can change the flow of the project between sprints. Similiarly, multi-team agile projects need to consider team management and team communication, as well as using distinct architectures and designs.

CCS Concepts • **Software and its engineering → Software creation and management**; *Software development process management*; Software development methods; Software development methods.

Keywords agile, teamwork, design, architecture, software development process

ACM Reference Format:
Cecilia La Place and Chiranjeevi Ramamurthy. 2018. Enabling Agile Through Teamwork. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1122445.1122456

1 Introduction

Agile development considers an element crucial to all software processes in a uniquely different manner: customers as team members. The basis of all software development is planning out a project from start to end, but requirements are prone to change when considering the fluctuating world of technology. Agile handles change by valuing the people involved, customers and developers [5]. By interfacing with customers frequently, and considering their changing requests as the project progresses, change is a byproduct. However, the success of Agile is reliant on what Cockburn and Highsmith call "responsive people and organizations" as well as "[focusing] on the talents and skills of individuals" [4]. As a result, teamwork and communication become unavoidably ensconsed in the agile process. In this work, we first, introduce a quick overview of Agile and agile teams in order to discuss multi-team projects in agile. Second, we discuss the different architectures used alongside these projects and design considerations.

2 Teams in Agile

A single team in agile consists of the product owner, the scrum master, and the developers. In order for their team to be successful, they must "have a common focus, mutual trust, and respect," be "collarborative, but speedy [in their] decision-making process" and be adept at handling ambiguity [4]. The Agile Manifesto reinforces these attributes by valuing customer collaboration and responding to change over contracts and plans [7]. Daily stand-ups incite communication, updating each other on task statuses, and conveying problems if they exist. Depending on the needs that follow the stand-up, the team will respond accordingly. e.g, team members may pair up to get a difficult or time sensitive task done.

2.1 Scaling Up Teams in Agile

As projects scale up, so does the number of teams. When considering this in an agile workspace, there is more work that must go into ensuring that teams themselves can collaborate effectively. If teams are in the same location, then promoting a collaborative environment is key. However, teams across more than one location must account for different cultures and providing as much possibility for communication as possible [6]. Distance increases the potential for misunderstandings. Other recommendations include having a single product backlog [11], cross-team daily stand-ups (and other cross-team activities) [10], and having communities of practice [1]. Each of these suggestions tackles a unique but very real and potential problem that occurs in large scale agile practices. Having a single product backlog allows all teams to see the whole picture, and see where work is needed. Cross-team activities allows for a deeper understanding of the project and each team's accomplishments and troubles.

*Both authors contributed equally to this research.

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

Finally, communities of practices creates a valuable way to share information about relevant topics from other perspectives.

3 Design and Architecture in Agile

3.1 Architecture

Constraints such as financial, regulatory, technical, or customer driven are some of the fundamental reasons for the failure of a well established organization if their agile process is not bounded by a fail-safe plan. In order to mitigate these issues, there is an emerging trend which explores the possibility of using Agile practices in order to manage traditional business functions. [9] A particular scenario where this issue is well addressed is the Scaled Agile Framework (SAFe).

The SAFe introduces two distinct elements of Architecture critical to its success:

- `Emergent Design`
- `Intentional Architecture`

The technical basis for development and the incremental implementation of initiatives are provided by *Emergent Design*. To ensure the initiative continually delivers value, Emergent Design helps Designers and Architects to be highly responsive to ever-changing customers and stakeholders. At this juncture, architecture that is SAFe can be seen as a collaborative and interactive exercise through which the design element can emerge.

Intentional Architecture however, is a more traditional architecture where the performance and usability of the initiative is supported and enhanced by a set of well-defined and planned architectural initiatives. The severity and importance of the constraints, such as choice of technology platform, financial budget, etc is clearly visualized using this Intentional architecture.

The key to the success is the level of abstraction at which the balance of Emergent Design and Intentional Architecture occur.

Agile requires architecture that supports how Agile practices deliver outcomes (value). This is achieved through a combination of a nimble reactive style of architecture supported by a more traditional and structured approach to rchitecture [3]

3.2 Design

In large teams, and multi-team projects, design plays a significant role. Design is often the center of a project, and finding the right design can be difficult. There is a risk of a project adapting to other development approaches if the initial step of "finding the right design" fails. Therefore, effective integration of design is needed. This kind of team integration is possible only if every single person in the team has a deep insight about the product and has prior experience. Extra work could be generated if a designer does not work closely with their team, thus leading to harmful silos of knowledge within the team(s). [2]

Avoiding this is crucial when a cross-functional team is involved in developing a product for intended customers/shareholders. By including designers in the agile planning process and pairing them with a developer to share their knowledge, this detrimental problem can be avoided. Having a visible work flow across the entire project is also helpful for all to see and work from. This approach enables a collaborative view, rather than a traditional linear view, and encourages the flow of ideas.

4 Conclusion

In conclusion, agile projects with multiple teams are heavily dependent on communication, collaborative cultures, and extra information resources. When scaling an agile team, it is absolutely necessary to ensure that the ongoing scaling process facilitates the scaling of the architecture. In terms of design, it is mandatory that designers have access to developers, and designs are visible to the entire team(s), in order to ensure project success.

References

[1] Heidi K. Gardner Alia Crocker, Rob Cross. 2018. How to Make Sure Agile Teams Can Work Together. Retrieved January 20, 2019 from https://hbr.org/2018/05/how-to-make-sure-agile-teams-can-work-together

[2] Scot W Ambler. [n. d.]. Agile Design. Retrieved January 20, 2019 from https://www.agilemodeling.com/essays/agileDesign.htm

[3] Scott Comte. [n. d.]. Does Agile need Architecture to be successful? Retrieved January 20, 2019 from http://enterprisearchitects.com/does-agile-need-architecture-to-be-successful/

[4] Alistair Cockburn Jim Highsmith. 2001. Agile Software Development: The People Factor. *IEEE Computer* 34, 11 (Nov. 2001), 131–133. https://doi.org/10.1109/2.963450

[5] Jim Highsmith and Alistair Cockburn. 2001. Agile Software Development: The Business of Innovation. *IEEE Computer* 34, 9 (Sept. 2001), 120–127. https://doi.org/10.1109/2.947100

[6] Sandeep Joshi. 2012. Agile Development - Working with Agile in a Distributed Team Environment. Retrieved January 20, 2019 from https://msdn.microsoft.com/en-us/magazine/hh771057.aspx

[7] Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas Kent Beck, Mike Beedle. 2001. Manifesto for Agile Software Development. Retrieved January 18, 2019 from http://www.agilemanifesto.org/

[8] Sven Peters. [n. d.]. Agile Design-Process and guidelines for collaborative design. Retrieved January 20, 2019 from https://www.atlassian.com/agile/design

[9] Carlos Pliego. 2018. Software Architecture: In an agile environment. Retrieved January 20, 2019 from https://medium.com/@carlospliego/agile-software-development-and-architecture-3dbb243fcbd2

[10] GSA Tech. [n. d.]. Collaboration Across Agile Teams. Retrieved January 20, 2019 from https://tech.gsa.gov/guides/Collaboration_Across_Agile_Teams/

[11] Sanjay Zalavadia. 2016. Techniques to scale agile across project teams and organizations. Retrieved January 18, 2019 from https://www.getzephyr.com/insights/techniques-scale-agile-across-project-teams-and-organizations