

Dealing with Large Solutions using ‘Scaled Agile Approach’: A Practitioners Slant

Dr. Jitesh R. Neve¹

Sr. Software Analyst, CloudMoyo India Pvt Ltd, Pune, India¹

Abstract: As a process setting practitioner of software process, many times I came across some disputes. Sometimes our patron party (client) not able to explicate what precisely he is expecting. Maybe sometimes people who faces customer (business analyst) fail to articulate requirements, which is fairly parallel to customer requirement. We understand that definitely there are some problems at both the ends like- novice user, lack of solid visualization power of business analyst, less skilled people, unstable requirements, etc. If we do root cause analysis, we’ll find ‘Agility’, which fits to incremental and rapid change of software requirements and technology under development. Recently, in any IT industry, it is not that stress-free stuff to carry on delivering solutions and meet ever-changing requirements of client without a suitable plan. Commonly, scrum practice is applicable in majority of IT organizations who clutch client centric culture. Going one level up, involvement of more than one (cross track) teams complicates monitoring deliveries and leads to dependency. So Scaled Agile is applicable at the level of ‘Program Management’. When we think of a large solution, we would need a more refined model to mitigate the risks as well as challenges. Right now there is no distinguished model exists which will be expedient specifically for Large Solutions at the program management level. I have tried to put the ‘Scaled Agile’ framework especially suitable for ‘Large Solutions’ through this paper. I have tried to present a practitioners point of view, especially biased on the productivity therefore the paper titled as ‘slant’.

Keywords: Agility; Daily Scrum Meeting; SAFe; Large Solution SAFe; Scaled-Agile; Scrum; Agile Approaches; Sprint; Lean; DevOps; PI Planning; Release Train; IP iterations; Large Solution; Solution Train

I. INTRODUCTION

To start with the introduction of this paper, first of all making clear understanding of ‘Agile’ and ‘Scrum’ is important. Let’s start with ‘Agile’; the dictionary meaning of this is, ‘ability of a quick as well as easy movement’. In other hand, development process of an efficient software comprises of certain set of rules and inter-dependent stages. This way fitting ‘Agility’ to a software development process is called as ‘Agile Model’, ‘Agility’, ‘Agile Approach’, etc.

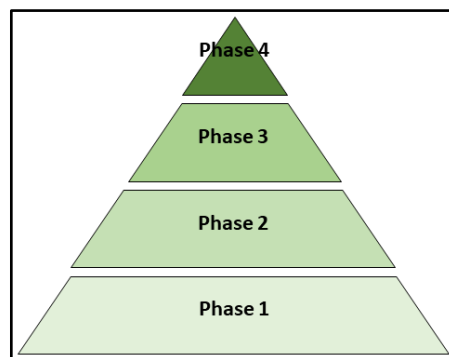


Fig. 1 Non Agile Process

Now a ‘Scrum’ word came here from a popular English game ‘Rugby’. In this game ‘Rugby’ stands for - restarting play right after a little invasion/infringement. In terms of software development, it means an iteration or a repetition. So collectively if we look at the scrum based agile model, we’ll get to know that Agile is nothing but an iterative software development model. Unlike other non-agile models, after every iteration (i.e. at the end of every sprint) there will be a working deliverable. Let me explain this thing with an example. Just for an instance, assume that we want to build a pyramid in the region of bad climate where anytime we have to stop working. But there are some conditions. First is, it is must to have a pyramid; otherwise the efforts and time invested will be lost. And second is we can place newly framed part once in a week. Now, see fig. 1; it describes building pyramid in non-agile way. Here if we abort construction in between, it will definitely not be a pyramid.

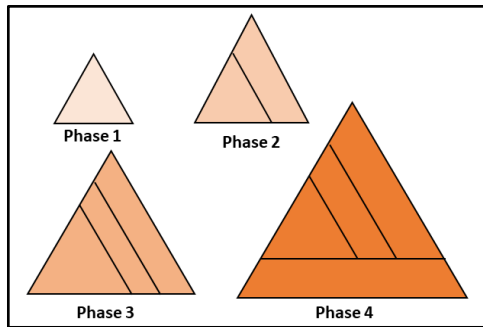


Fig. 2 Agile Process

In fig. 2; you can see a pyramid building process by agile mode. In this way, at the end of every phase there will be a pyramid. In case, for some reasons we stop working on it, we will get a required thing. Same thing happens with Software, at the end of every sprint there is a working solution delivered to the customer. This enhances the user experience as sticks to legal and business compliance [1, 2 and 3].

II. THE AGILE MODEL

From a perspective of a process setter, when we observe a perfect looking agile model does not look same. This differs from organization to organization or may be team to team; as the culture and followers change. Some of them follow this just for a compliance. Or some client centric (typically service based) organizations use this just to product delivery and earn bonus/profit. Some of those believe on backlogs only. Quite a few follows this seriously and trust on retrospections followed by improvements [1, 2].

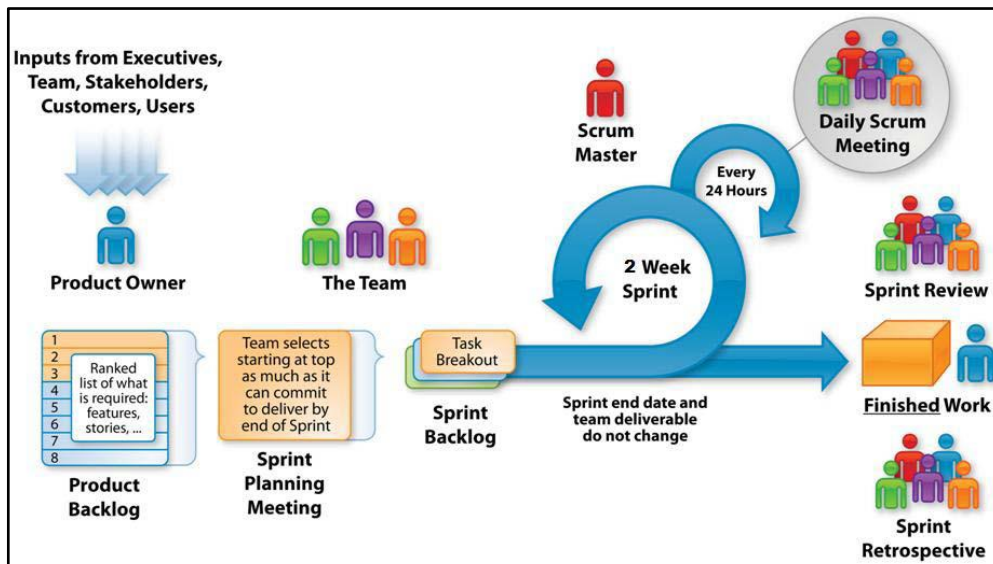


Fig. 3 A typical agile model

Fig. 3 shows a typical agile model. Where user stories/requirements are considered as backlogs. Product owner (sometimes client) defines the requirements. During sprint planning Product Manager involved to prioritize and distribute the backlogs. Team (developer and tester both) works on assigned backlogs during sprint. One scrum master is there for assisting and monitoring the work-items assigned. During sprints daily scrum meeting (usually called as standup meetings) happens to discuss the work done yesterday, what is to be done today and dependency if any. At the end of every sprint, new build is deployed. After every sprint, sprint review and/or sprint retrospective meetings are conducted for improvements [2, 3].

III. CHALLENGES IN AGILE

- For large projects, many teams might involve during development/testing the product. Like – DB team, UI team, team responsible for writing data integrations, reports development team, API team, etc. every team may have its own sprint/process. So monitoring at the program level becomes moderately difficult.
- There is no provision in agile for cross-team interactions.



- Agile is more suitable for a single team.
- Daily standup meetings may take a long and time consuming if someone started a lengthy unnecessary discussion on some point of understanding which may not be relevant for whole team.
- Lack of maturity of team members may increase some dependency. It induces unnecessary escalations and figure pointing.

A. *Customer Experience*

Sometimes client organizations has massive potential but they fail to expose how processes can be utilized? To perform quicker and effective way, disputes usually begins if –

- Customer / product manager unable to prioritize
- When client or requirement provider is unaware about agile
- Or maybe he is not wise enough to realize that not everybody will work on every platform for full time.
- When he doesn't aware that change in requirements might waste efforts of the team members.
- When planned/expected work-items not delivered in committed timelines.
- Unnecessary escalations and finger pointing affects bandwidth of people.

B. *Impact on Revenue*

There is no second opinion, delayed delivery makes customer unhappy and attracts fine. This is a major concern of organizations. So higher management always promote prompt or before time delivery to claim a bonus to customer. Apart from billing loss organizations customer engagement may fall. This may impact on NPS (Net Promoter Survey) score. Ultimately organization may lose future business opportunities with the same customer engagement [1, 3].

IV. SCALED AGILE FRAMEWORK (SAFE)

The SAFE (Scaled Agile Framework) supports business and considerable difficulties during development to delivery of the excellent software systems in minimum bearable lead time. We can call it as one type of predictable success patterns for executing Lean-Agile software at enterprise scale. 'SAFE' is especially designed for program management level visibility. In this model we can see many challenges of normal agile methodology are been disappeared. Here we can easily deal across the teams.

There are 4 types of SAFE framework, 'Essential SAFE', 'Portfolio SAFE', 'Large Solution SAFE' and 'Full SAFE'. Many people believe 'Essential SAFE' framework is the fully equipped scaled agile framework. But in reality 'Essential SAFE' is a fundamental pattern of SAFE.

V. LARGE SOLUTION SAFE

When we think about large solutions, we need to have a more refined model to mitigate the risks and challenges. Right now there is no distinguished model exists which will suit especially for Large Solutions at the level of program management.

You can see fig. 4 for the large solution SAFE configuration. The configuration is intuitive enough to understand. The whole configuration has been divided into 3 sections, namely – Team, Program and Large Solution.

Team section showing involvement of more than one teams working concurrently on different streams of large application. Each team can have its own process like – scrum/agile/Kanban. People involved are team members, scrum master, product owner. Processes involved are planning, execution, review, retrospection, code submission.

Program section is the middle section of this configuration. Here, program manager has strong visibility over the team section. It contains continuous delivery pipeline. Agile release train is the process runs over the pipeline. Whenever there is a demand for release/deployment, the train is initiated. Once the train is started, it runs collecting all code submitted to repository for every module/project; builds and deploy it. Processes involved are system architecture, product management, CE-CI-CD (continuous exploration-continuous integration-continuous deployment), automation, recovery and lean flow.

Large solution is the top section of the configuration. This is the customer facing layer. Here solution architect and solution management is involved to control the solution configuration. Large solution section is responsible to deliver business solution to the customer. In addition, it includes processes of solution compliance management to comply with legal/auditing entities. Economic framework also comes to picture for the budget management of the large solution



project. It contains solution train, it usually collects the specified build deployed by the agile release train and delivers the sanctioned build to the end customer. Pre and post Solution demo is also included in this section.

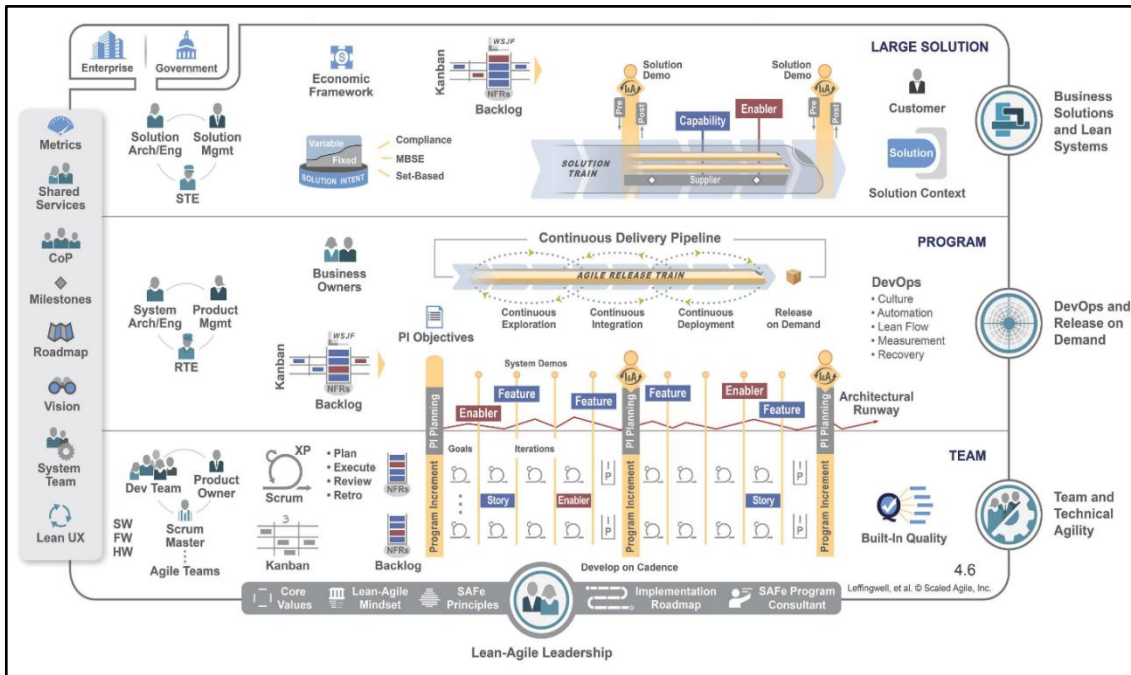


Fig. 4 Large solution SAFe configuration

VI. BENEFITS



Fig. 5 Business benefits

See fig.5 for business benefits. I think, its more intuitive visual. In SAFe method, refining business outcomes for almost every scale of companies across the globe. It has successfully shaped dramatic growths within timeline to sophisticated quality, market, employee as well as customer engagement, and finally (most important) overall value-added commercial conclusions. It is proven that creates cultures those are more gratifying, fruitful and exciting.

VII. SUCCESS FACTORS

See fig. 6 for success factors of agility. Apart from this, there are numerous factors those may affect productivity success of Scaled Agile. According to my experience below are the said factors whose impact will be quite severe.

- Less market knowledge/analysis
- Lower team morale
- Being distracted from short term goals
- Changing/improper priorities
- Non-cooperation among people
- Inappropriate management of customer expectations
- Lack of time management skills
- Lack of commitments

- Poor management of immature resources
- Frequent changes in 'nature of business'

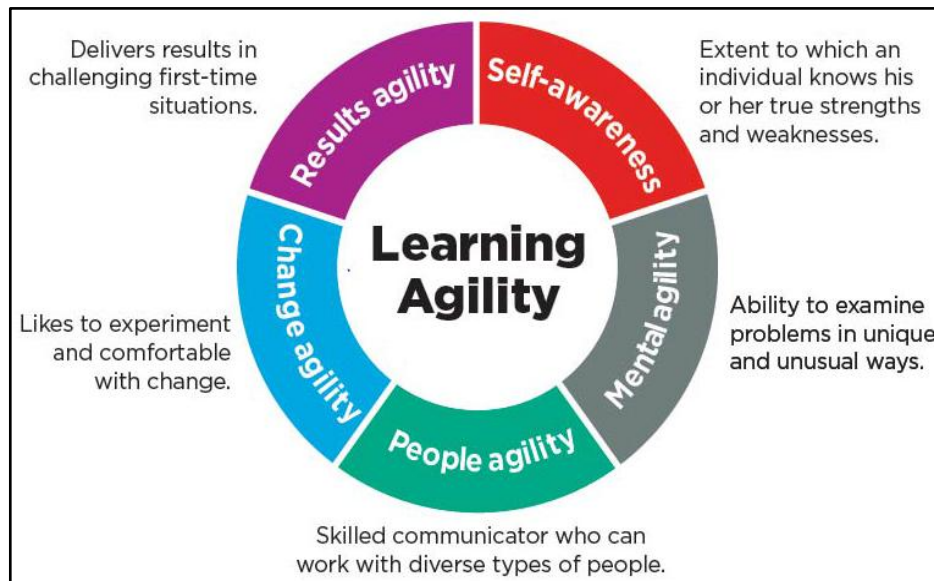


Fig. 6 Success factors of agility

VIII. CONCLUSION

As a process setting practitioner, I believe for the improvement in the process and productivity along with dealing with large scale solution, there are incredible reviews and retrospections going on worldwide. Hence, dealing with large solutions, along with scaled agile approach will fit in the concepts written in this paper. I am sure that improvement in productivity will also improve business benefits. It has been seen that evolution in process leads to the better economic outcomes for every organization. Before this paper there was no experimented scientific writing of any model existed which will be convenient explicitly for Large Solutions. I have tried to present a practitioners point of view, especially biased on the productivity therefore the paper titled as 'slant'.

ACKNOWLEDGMENT

This research article is dedicated to the Supreme Lord 'Krishna' and my loving parents. I would also like to expose my gratitude towards my managers and project team members, who privileged us to implement, experiment and monitor the written process. Special thanks to **Kaustubh Vaze** and **Avinash Sawant** for his necessary support.

REFERENCES

- [1]. Dr. Jitesh R. Neve, Kamini Godbole and Rajeshwari Neve, "Productivity And Process Improvement Using 'Scaled Agile' Approaches: An Emphasized Analysis," 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, 2017, pp. 793-798, doi: 10.1109/ICICI.2017.8365245, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8365245&isnumber=8365179>
- [2]. S. P. Patil and Dr. Jitesh R. Neve, "Productivity Improvement of Software Development Process Through Scrumban: A Practitioner's Approach," 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, 2018, pp. 314-318, doi: 10.1109/ICACCT.2018.8529405, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8529405&isnumber=8529319>
- [3]. Kamini Godbole and Dr. Jitesh Neve, "Kanban: A Definite Answer for Effective Project Execution in Uncertainties," in International Journal of Advanced Research in Computer and Communication Engineering (IJARCCCE), Vol. 6, Issue - 6, pp. 561-565, June 2017, doi: DOI10.17148/IJARCCCE.2017.66102, URL: <https://ijarccce.com/upload/2017/june-17/IJARCCCE%20102.pdf>
- [4]. Torgeir Dingsøy and Nils Brede Moe, "Towards Principles of Large-Scale Agile Development: A Summary of the Workshop at XP2014 and a Revised Research Agenda", in Pringer Journal XP 2014 Workshops, LNBP 199, pp. 1-8, 2014.
- [5]. Aditya Pancholi and Sapna Grover, "Scaled Agile Framework: A Blight", in International Journal of Innovative Research and Development (IJIRD), Vol. 3, issue - 5, pp. 425-427, May 2014.
- [6]. Igor Stojanov, Masters Thesis on "Scaling Agile using Scaled Agile Framework", Department of Mathematics and Computer Science, Eindhoven University of Technology, Netherlands on Aug 2014. Available online at - <https://pure.tue.nl/ws/files/46987782/782896-1.pdf>.
- [7]. Torgeir Dingsøy and Nils Brede Moe, "Research Challenges in Large-Scale Agile Software Development", DOI:10.1145/2507288.2507322, in "ACM SIGSOFT Software Engineering Notes", Vol. 38, No. 5, pp -38 and 39, September 2013.