Agile Software Development- What works, What may not & What could help

Rahul Agarwal

Abstract— In February 2001 seventeen software developers brainstormed and defined the core principles of Agile Software Development methods which is better known as the Agile Manifesto. Supplementing the manifesto, the Twelve Principles further decide what it really means to be Agile. This paper takes into cognizance those twelve renowned agile principles and how we can leverage each to best use. The arguments emanate from practical exposure of leading large number of agile teams (18 Scrum teams supported by Core team and Cross Functional teams at the peak) working towards a common goal.

Index Terms— Agile Software Development (ASD), Unique Selling Proposition (USP), Subject Matter Expert (SME), Product Owner (PO), Cross Functional Team (CFT), Business Analyst (BA), Definition of Done (DoD).

1 Introduction

Agile Software Development (ASD) is the talk of the town. Organizations, irrespective of size, take pride in associating themselves with anything related to Agile. It's become sort of status symbol for organizations whether or not they are following Agile Software development. Some organizations even go the extent of quoting ASD as their USP.

Although there are success stories written all over about Agile, mere introduction of Agile will not guarantee success. It has to be an elaborate, thought-through process which needs Senior Management as well as customer buy-in.

Even though I am sold on Agile (having been working for last two years in an out and out Agile based engagement) and have personally experienced tremendous gains we have to be optimally cautious while inducting Agile better practices/Scrum methodology in an existing program or starting a new program on pure play Agile framework.

2 The 12 Principles of Agile

1. *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*

   What works- Deliveries are made in each sprint cycle at regular and defined intervals. This inevitably leads to sustained Customer focus and hence increased Customer satisfaction.

   What may not work- Customers get used to continuous deliveries thus bearing large expectations even in case of insufficient backlog/grooming when teams might not be in position to make delivery. Customer sometimes gets more creative by demanding frequent deliveries during the sprints, causing pressure on Scrum Team and thus breaking Agile principles.

   What could help- Set clear expectations with the customer. It would be all the more better if they are penned in the contract to avoid any unforeseen situation which could hurt monetarily. Having frequent and healthy communication with Customer regarding impediments/distractions helps running effective Scrum Practice and effectively delivering high quaility software to Customers.

2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*

   What works- Prioritization of requirements (or user stories) by Product owner/SME. Development completion, even for changing requirements, is much faster as compared to traditional SDLC methods.

   What may not work- Requirements (read Backlog) come late during the sprint cycle and teams are supposed to make delivery within same cycle.

   What could help- Do welcome changing requirements but before the start of the sprint cycle and also ensure to set the priorities right for those.

3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*

   What works- Well defined & religiously followed Sprint cycles (usually 2 weeks) and commitment from

• *Author (Rahul Agarwal) is currently pursuing Ph.D. in Management from Amity University, India. E-mail: rahul1agarwal@gmail.com*

both customer as well as scrum team towards phased, regular as well as demonstrate-able stories/deliveries.

What may not work- Team members tend to burn out because of the recurring nature of sprint cycles. Due to no break in between people are found gasping for breath after 3-4 sprint cycles.

What could help- Have an empty sprint in between which could help team members rejuvenate and prepare well for forthcoming sprints. This empty sprint need not be 'empty' per se- it could be utilized for bug fixing/regression testing etc.

4. *Business people and developers must work together daily throughout the project*

What works- Team composition- Business Analysts as part of Scrum teams.

What may not work-1 BA per team even in Support & Maintenance kind of work might be an overkill.

What could help- A Business Analyst can be divided across teams should there be an engagement which is high on Support & Maintenance component.

5. *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done*

What works- Teams deciding the stories to be taken up in a sprint cycle- empowerment of ground workers in real sense.

What may not work- Team members getting routine tasks for prolonged time.

What could help- Scrum Master should not play the role of a pure play Project Manager. She/He should enable/facilitate the team members to take decision when it comes to sprint cycle related delivery.

6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation*

What works- Daily Stand-up meeting- it helps Group Scrum Master/Scrum Master/Team members track the progress and highlight & resolve impediments faced by the team.

What may not work- Team members waiting for the daily stand-up meeting to get an issue resolved.

What could help- It always helps to keep the stand-up as brief as possible and just discuss on 3 items- progress, impediments and action.

7. *Working software is the primary measure of progress*

What works- Client gains confidence steadily as teams/Product owners/SMEs demonstrate a working copy of software with incremental gains over past sprint.

What may not work-Software works but there are bugs, especially high severity bugs which hinder functionality.

What could help- Have a crisp Definition of Done with clear technical as well as functional understanding of the backlog/stories to be developed during the sprint.

8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely*

What works- Mutual trust and understanding gained during the course of development leading to better planning and hence better delivery.

What may not work-The customer might want to run faster; Regular sprint cycles without a break drain team's energy as well as enthusiasm.

What could help- Customer representative/SMEs/Cross Functional team/Scrum team should be on the same page right at the start during sprint planning phase itself.

9. *Continuous attention to technical excellence and good design enhances agility*

What works- Senior Specialists within Core team and Cross Functional teams, Regular technical reviews, In-scrum QA team members.

What may not work- Architecture and Design, once frozen, can't be changed unless absolutely essential.

What could help- Having senior technical folks, even if some of them are not billable; regular design reviews keeping in cognizance evolving requirements.

10. *Simplicity--the art of maximizing the amount of work not done--is essential*

What works- Keeping the tasks simple and elaborate; having absolutely crisp stories which means having a very neat backlog creation.

What may not work- Unclear requirements, Hazy Definition of Done criteria.

What could help- Clearly defined roles within Scrum teams (Developer/Tester/Business Analyst), Absolute clarity on Definition of Done, Granular stories for the team to work upon.

11. *The best architectures, requirements, and designs emerge from self-organizing teams*

What works- Empowered teams.

What may not work-Lack of Senior Technical and Functional people in Scrum teams.

What could help- Have a good mix of Functional as well as Technical people both within Scrum team and as contributing members of Core and Cross Functional teams.

12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly*

What works- Retrospective discussions held after completion of each sprint.

What may not work- Team members not taking retrospective in right spirit.

What could help- Retrospective action items should be listed exhaustively and tracked meticulously.

3 Conclusion

Agile Development is both practical as well as fun. Propagated/Exercised/Implemented with passion, vigour and care it has the potential to yield surprising results.

For Program/Project success it's not necessary to follow the 12 Agile principles to the hilt. We just have to be prudent to analyse which principles suit our business needs, carefully chalk out the scenarios, obtain complete management/customer buy-in and then embark the Agile journey.

.

References

[1] http://agilemanifesto.org/principles.html

[2] http://scrummethodology.com/the-agile-manifesto-and-twelve-principles