



An Omni Sourcing White Paper
June 2017

Can Enterprise Organizations Truly be Agile?



Table of Contents

Introduction..... iii

What is Agile?..... 1

Do the core values of Agile make the best fit for enterprise organizations? 2

 Working software over comprehensive documentation..... 2

 Responding to change over following a plan..... 3

 Individuals and interactions over processes and tools..... 3

 Burden of subcontracted organizations and time zones 4

 Customer collaboration over contract negotiation 4

Conclusion 5

Introduction

Over the years, there was a reaction to traditional methodologies – particularly waterfall – and thus the movement towards alternatives to traditional development. The objective was to help teams better respond to unpredictability through incremental and iterative processes and empirical feedback.

This movement created Agile. Agile has become an appealing alternative to software professionals because it is quick and you can iterate as you move along the process, so companies striving to improve their performance get on board with Agile. The perception is that age old IT problems will get solved with Agile: teams will become effective, true collaboration will occur, customer feedback will be established, resulting in improved product quality and incorporated customer feedback.

It is challenging to criticize Agile for fear of being labelled “old school”, however, not every business need for software development/testing can be solved by pure Agile methodology. Enterprise organizations need to know this. According to Boehm and Turner 2005, the methodology was originally designed for small and individual teams focused on single-team projects.

This white paper defines Agile methodology, evaluates its core principles and describes the critical reasons why the absolute use of the methodology is not recommended in enterprise organizations with distributed teams. Our sample size comprises an average of 11 enterprise organizations. Each organization has software development/testing teams working out of at least 2 of the following continents: North America, Asia, Europe and Africa. The paper addresses the following questions:

- What is Agile?
- Do the core values of Agile make it a best fit for enterprise organizations?
 - Do enterprise organizations really need to use agile methodology in software development/testing?
 - Is pure agile methodology the best practice for enterprise organizations?

What is Agile?

Agile is an approach to project management that strives to unite teams around the principles of collaboration, flexibility and responsiveness to feedback. Agile testing is the practice of testing software for bugs or performance issues within the context of an Agile workflow (Sprint).

The Agile manifesto as shown in Figure 1 resulted from uncovering better ways of developing software by doing it and helping others do it.

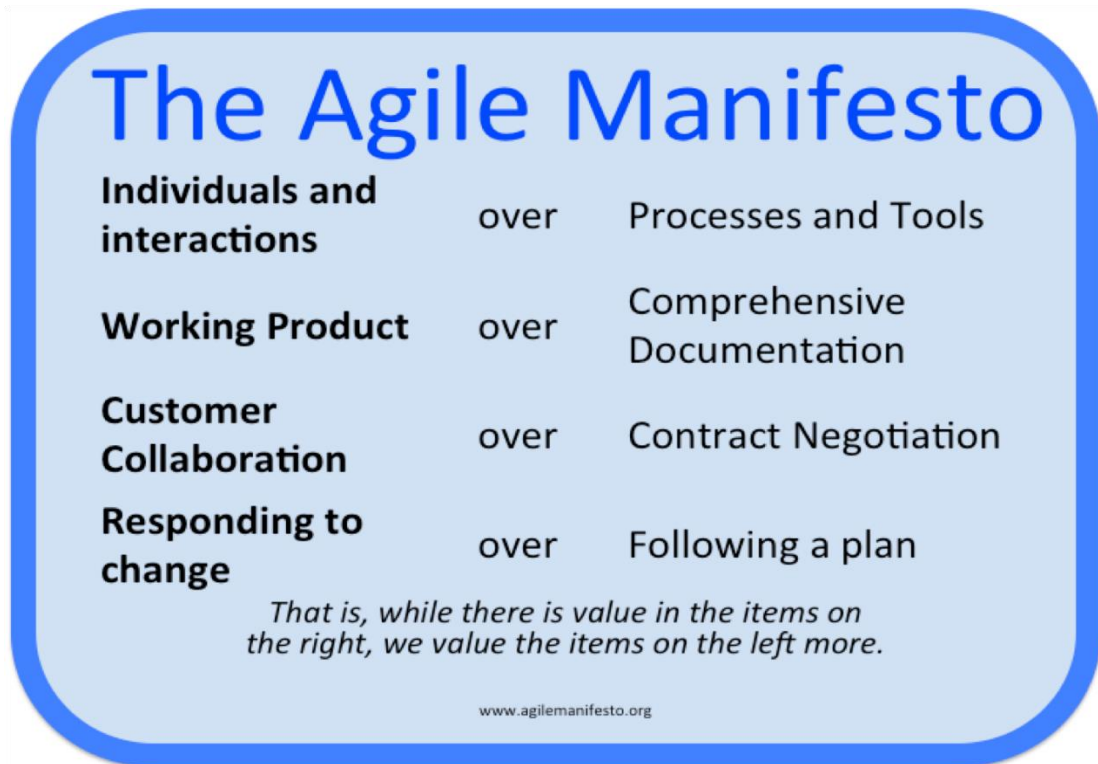


Figure 1: The Agile Manifesto

Agile proponents clearly state that there is value in the things on the right. The problem is that because the methodology came by – and it promised delivery of results, which it does greatly for small to medium organizations – most enterprise organizations were quick to jump ship without putting much thought into how it can be adapted to suit the uniqueness of each organization.

There is a great contrast between what Agile proponents stated and what enterprise organizations that are adopting agile are practicing. There is no need to do a dump of the items on the right because you want to be identified as becoming “Agile”. There is nothing wrong with embracing the items on the right.

A key emphasis of Agile is that face-to-face, spontaneous conversation is the best form of communication. This means co-location. The argument is that teams can use technology to maintain open and continuous communication. We do know that organic discussions are generated in open plan offices which Agile emphasizes. These discussions are spontaneous. Technology – open phone lines, meeting room via IMs – has not yet caught up with this spontaneity. A developer who is far away from

the office where the spontaneous discussion ensued is only going to hear the second-hand version i.e. the perceived way someone heard that discussion.

Do the core values of Agile make the best fit for enterprise organizations?

Agile methodology or movement, if you like, requires an IT organization to practice the manifesto. However, the core principles that Agile emphasizes are the main reasons why it is not best practice for enterprise organizations.

Working software over comprehensive documentation

One significant difference between small organizations and large organizations is that large organizations have more dependencies between projects and teams. These dependencies are really hard to adhere to in delivering a testable feature or component that can be tested after each sprint. According to Lindvall et al., 2004, this increases the need for formal documentation and thus reduces agility.

While Agile may help you churn out releases faster, your employees may struggle with maintaining the software when there is little documentation to utilize in supporting/testing the product. What good is a software that cannot be maintained? What if the employee who developed the product suddenly drops off the surface of the earth? Then you will have a team scrambling to figure out how things work. There has to be documentation for succession planning.

Yes, we do want working software; however, we need documentation to go along with that software as well. It is better to create and follow a plan – Agile is not excited about plans and we'll get into that shortly – whereby there is a buffer for the developer or the tester to produce agile documentation on the software developed.



Figure 2: Document with agility

Responding to change over following a plan

Yes, Agile is not particularly excited about plans. Agile emphasizes incorporating changes instead. If we keep responding to changes at every phase – requirements, development and testing – without having a plan in place for handling change requests, is the team going to ever finish developing the product? What if people request for big things late in the project under the impression of “it’s an agile project and we should be able to handle it because we respond to changes”? Now, don’t get it wrong, change is good, after all that is the only thing that is constant but, before we make changes, we should ask ourselves the following questions:

- Why was the change not initially scoped with the project?
- Why is it happening now?
- What is the impact if we do (not) make the change?

If these questions have been answered satisfactorily, then let’s go ahead and make the change.

Enterprise organizations need to understand that they, above all other types of organizations, are the ones that need to follow a plan because they are large and there will definitely be many proverbial cooks in the kitchen on any project. Lack of planning should be a concern as business and customer relationships build on a long term roadmap. A plan helps keep the project on track because you have a guideline to refer to and ensuring that things don’t get messy along the way. Focusing on continuous delivery typically has the effect of creating an unmanageable backlog of defects because developers are constantly working to deliver software.



Figure 3: The purpose of planning

Individuals and interactions over processes and tools

Large enterprises often comprise big distributed teams. It’s important to Agile to collaborate so if distributed teams are not able to easily communicate via technology tools, then, it will be difficult to contribute. Leading Agile CEO, Mike Cottmeyer, noted in an interview with StickyMinds, that “at scale in

larger organizations, it's incredibly difficult to find a pattern that allows teams to be able to come together and stay together and be held accountable and establish stable velocity and all that stuff".

Burden of subcontracted organizations and time zones

Another common occurrence with enterprise organizations is teams composed of members from various subcontracted organizations. These types of team typically lack collaboration. The team comes from different organizations who may not share the intrinsic goals and objectives of the partner organization or even other vendors that they are required to work with. There can be conflicting project objectives within this type of team. Time zones can also be an inhibitor to efficiency and collaboration. Agile expects short iterations during which all members interact and actively collaborate and this is important to project success.

There needs to be a process in place for collaboration and communication within distributed teams and because they are not co-located, there is the need for tools to aid communication. Enterprise organizations need to be able to find the harmony between people, processes and tools because that's the only way they can create organizational agility.



Figure 4: Healthy balance between people, process and technology

Customer collaboration over contract negotiation

Early visibility is really important for the customer and feedback is great to have. It is definitely very important to keep the customer happy by talking to them. Yes, people matter, we get it. However, if a customer keeps coming back and adding scope or changes, when do you think you'll be able to draw the line? You would definitely have to refer to the agreement signed at the initiation of the project. Again, if requirement elicitation is done correctly, there will be less change requests.

**It's OK
to
Set
Boundaries**

Figure 5: Set boundaries

Conclusion

Agile is not a one-size-fits-all. It is not a panacea and doesn't solely fit enterprise organizations with distributed teams. If you have a company with a team in Asia and another team in North America (NA), more often than not, there is a wait time especially if you have to make rapid changes in NA and the person responsible for the change is in Asia. Also, innovative teams have flat structures and usually, this is not the case in large enterprises which is contrary to the principles of agile.

Agile is great for prototyping and iterative development but pure Agile is not recommended for large development projects. Enterprise organizations often deal with larger and more complex problems than small ones. Agile doesn't solve all management issues and large development projects are trying to manage communication, budget, scope, schedule, subcontracting companies, business units amongst others.

Our position is that sole use of agile methodology is not a best practice for enterprise organizations because we definitely value the items on the left as equally as the items on the right. Enterprise organizations that are undergoing agile transformation should learn from this. It will be more expedient for them to combine agile with another methodology – “Ag-Fall” – that defines the processes required to completely manage a project and rely on agile for day-to-day management of small co-located teams focused on a single project.

Omni Sourcing is focused on scaling Agile or combining it with other methodologies to ensure that you remain competitive in the marketplace. Our objective is to continue to provide customized solutions that effectively solve our clients' business challenges at a reduced cost without compromising on quality and performance.

To find out more about how your enterprise organization can benefit from effectively blending Agile and Waterfall methodology in large development projects/programs, visit omnisourcing.net/index.php/services/program-management-office/program-and-project-management or call 800-617-0200 for a free analysis.