



Agile Record

The Magazine for Agile Developers and Agile Testers



Complexity Thinking



View #1: Emergent Process
...ion, desires, understanding, relatedness

Management 3.0

Testing challenges for distributed team

February 2012

issue 9

www.agilerecord.com

free digital version

made in Germany

ISSN 2191-1320

Distributed Agile: The Maturity Curve

by Raja Bavani



Last month, I came across an interesting question from one of my friends in the industry. He asked, “How does a distributed agile team start on a project and make progress in delivering results?” I said, “It depends”, and paused for a while before discussing it at length. The next day at leisure I recollected a series of incidents from one of my previous agile projects and created a visual on how my team matured over a period of time. This project had multiple release cycles and hence ran over a period of three years. We had two teams – a team of 20 engineers in India and another team of 5 in the US. Right from our initial struggles

to adopt Agile I reflected on the way we matured. Also, I thought about what I would do differently if I were to start all over again. Let me share my thoughts and conclusions in this article.

In simple terms, adherence to the Agile Manifesto and Agile Principles is the essence of Agile. Agile teams choose either a popular methodology (e.g., Scrum) or put together a methodology that follows agile principles and practices. In reality it takes several months for agile teams (either collocated or distributed) to mature and reach the other end of the spectrum.

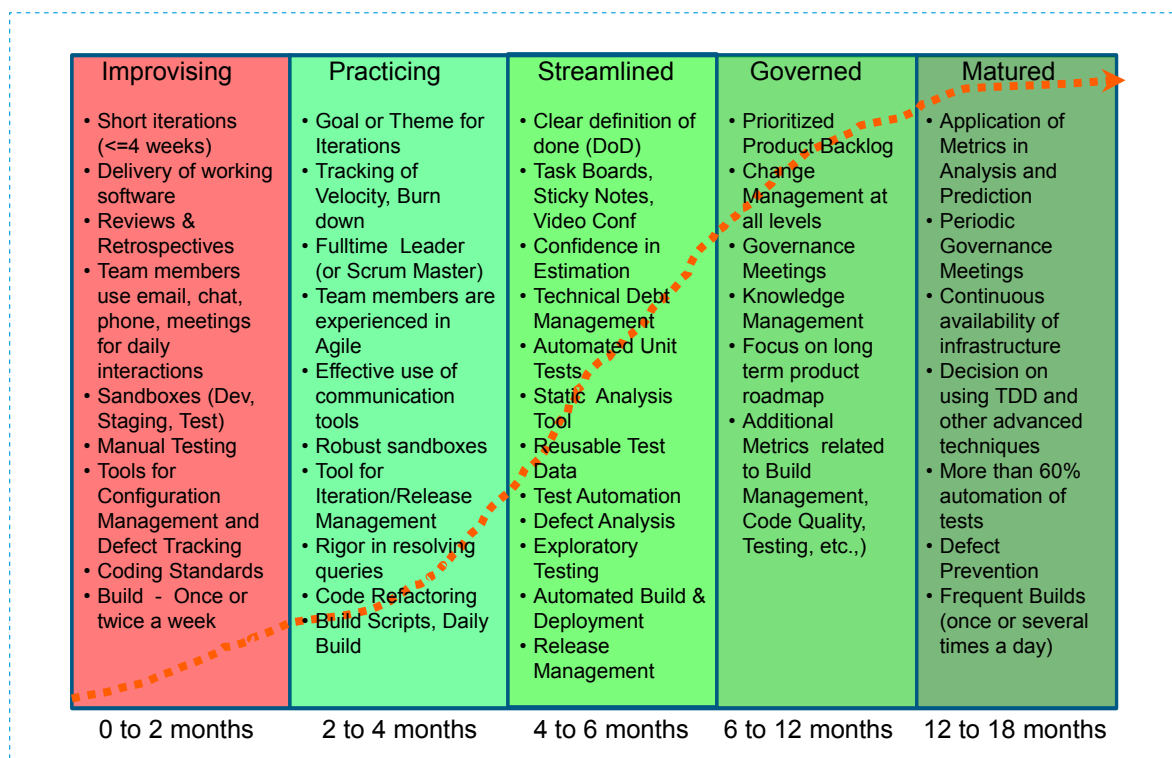


Figure-1 Maturity of Distributed Agile: The Past

Over the past eight years, I have seen many projects transitioning through these levels as shown in Fig-1, which is nothing but a refined form of the visual I created. In this figure, the duration specified for each level is the time taken from the beginning of a project or group of projects to reach that level. In distributed projects, because of multi-site communication and coordination, the first two months lapse sooner than expected. This is the crucial period that requires a lot of attention in making the right start in projects.

Looking back, my findings are:

1. The longer you stay at the first two levels, the worse it gets. Nowadays, 4 months is a longer time than it used to be.

2. If your comfort zone is “Improvising”, beware. Either your agility will dilute or your team and other stakeholders will express dissatisfaction because of your team’s inability to deliver results.
3. Moving through the second and third levels involves a steep curve. It is a complex affair in distributed agile. Experience and availability of experts can help you go through this tough journey.
4. Taking a long pause after reaching the third level is not a good idea. It is wise to push through level four and five as well. These levels are essential to deliver results in distributed agile projects.

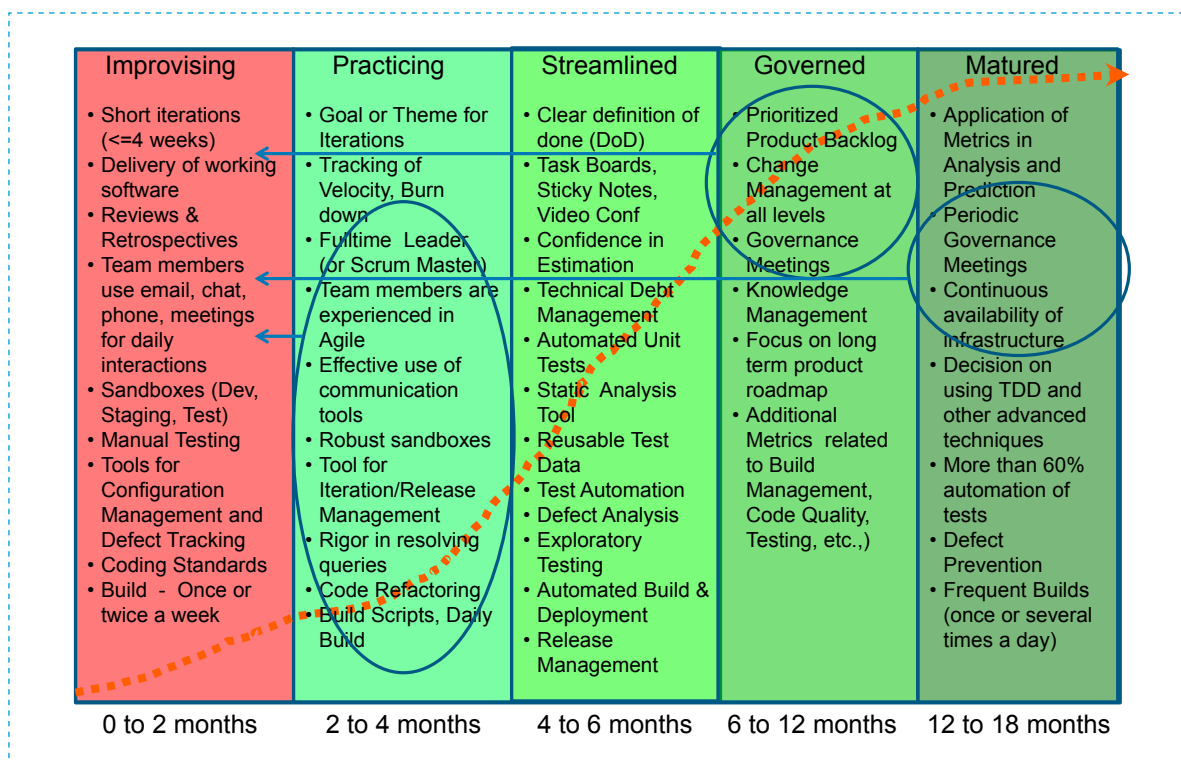


Figure-2 Maturity of Distributed Agile: The Present & Future

There are several ways to tweak these transitions in order to improve results during the early stages of agile projects. In a way, moving some of the items into the first level will provide us with immense benefits. Obviously, we need to be proactive in doing certain things early as shown in Fig-2. This is just the beginning and the following should be considered:

1. It is good to have ‘agile aware’ team members. However, it is better to have at least one or two team members who are experienced in agile. Besides, the availability and support of agile experts or coaches will have a positive impact on the team’s performance.
2. It is good to have email, chat, phone and other communication mechanisms. Effective use of communication tools is necessary to ensure efficiency. Agile teams cannot afford to use chat for lengthy conversations. They must know when to communicate over the phone.

3. It is good to have sandboxes (or environments) for development, staging and testing. However, it is necessary to ensure that the environments are robust.
4. Introducing a tool for iteration/release management is very important.
5. The build process needs to stabilize during the initial month.
6. Prioritized product backlog needs to be maintained from early stages.
7. Change management is essential. Otherwise the teams may not have a clear idea on how changes can be managed in practical situations.
8. There has to be a governance team (especially in case of distributed agile projects) with a commitment to have review meetings at regular intervals. Governance in distributed team is paramount for timely decision-making in vari-

ous areas that are outside the purview of the project team. A very good example is initiating and providing consistent support or sponsorship for the visits of team members at all levels across sites.

In distributed Agile, there are several factors that influence the ability of teams to become better in a reasonably short duration. Some of the key factors that consume significant efforts in communication and coordination include,

Refinement of user stories: When user stories consume significant effort to refine, Agile teams struggle to find adequate time to focus on implementing engineering practices in the right way.

Engineering practices: When there is no consistency or commonality across sites in implementing engineering practices such as unit testing, static analysis, continuous build, test automation, etc., teams will have to spend considerable time to make these work.

Technical debt management: When teams do not have a common understanding about managing technical debt, the situation will lead to severe technical risks in the project. Such risks will surface within the first three or four months from project initiation and hence will consume the efforts of team members in paying back technical debt rather than maturing existing practices.

Work standards: Differences in work standards mean differences in implementation of process and engineering disciplines across sites. This is the so-called 'double standard' phenomenon. In distributed agile teams, this is not acceptable and can lead to a situation where teams cannot work toward maturing their practices.

Long-running projects provide an opportunity to improvise, practice, streamline and mature over a period of time. This means an opportunity to maximize the benefits of continuous improvement. How do we execute projects that need to be completed in a short duration of 6 months or less? We can do this by being proactive as shown in Fig-2. Also, this means that projects need to rapidly transition from one end of the spectrum to the other. This is possible when teams exhibit an agile mindset, belong to an ecosystem that has experts who can coach agile teams, and have the necessary infrastructure to execute agile projects.

I am sure you are able to relate this discussion to your experience. In essence, the objective of this article is to emphasize that projects belong to a maturity spectrum or maturity curve. Project sponsors and stakeholders need to understand this aspect and encourage teams to move towards higher maturity levels in order to gain benefits from Agile. In distributed Agile, this becomes very complex. However, with experience and capabilities, one can demonstrate success.

> About the author



Raja Bavani

is Technical Director of MindTree's Product Engineering Services (PES) group and plays the role of product engineering evangelist and Agile coach. He has more than 20 years of experience in the IT industry and has published papers at international

conferences on topics related to code quality, distributed Agile, customer value management and software estimation. His areas of interests include global delivery model, Agile software development, requirement engineering, software architecture, software re-use, customer value management, knowledge management, and IT outsourcing. He is a member of IEEE and IEEE Computer Society. He regularly interfaces with educational institutions to offer guest lectures and writes for technical conferences. His product engineering blog is available at <http://www.mindtree.com/blogs/category/software-product-engineering>. His articles and white papers on Agile software development are available at: <http://mindtree.com/category/tags/agile>. He can be reached at raja_bavani@mindtree.com.