

Distributed Agile: Ten Guiding Principles

Raja Bavani, Chief Architect, Mindtree
28-Dec-2013

Abstract

This paper presents ten principles we learned through experience in working with project teams and our interactions with industry experts. These ten principles are above and beyond agile manifesto and twelve agile principles. These are related to areas such as context-specific methodology, tools for productivity improvement, infrastructure for communication and coordination, knowledge management, focus on quality, inclusion, collaborative governance, automation, technical debt management, iteration progression and ensuring early success.

Introduction

The challenges in distributed agile can be seen under three broad categories viz., a) Communication and Coordination, b) Time Zone Differences and c) Issues related to People, Culture and Leadership Style. Successful teams consciously adhere to certain principles and it is their principle-centered approach that helps them face such challenges and deliver the best. Steven Covey wrote, *“Principles always have natural consequences attached to them. There are positive consequences when we live in harmony with the principles. There are negative consequences when we ignore them. But because these principles apply to everyone, whether or not they are aware, this limitation is universal. And the more we know of correct principles, the greater is our personal freedom to act wisely.”* What he said is true in all situations of life and it includes application of agile methods in geographically distributed teams too.

Ten Guiding Principles

Simply put, adherence to Agile Manifesto and Agile Principles is the foundation of agile teams. In addition to the 12 principles declared by the signatories of Agile Manifesto, we value the following ten guiding principles. These ten principles are specific to geographically distributed teams that follow agile methods.

1. Methodology driven by project teams: Agile software development in a distributed environment does not mean step-by-step implementation of any specific agile methodology such as Scrum, with high expectations on on-time high-quality delivery. It means collaboration amongst distributed teams to collate processes that follow agile principles, and to put together a methodology that works for them. Projects that follow distributed agile suffer when a methodology accepted by a sub team drives the rest of the team. Successful distributed agile projects happen because of collaborative teams that drive to define a methodology for themselves. The definition of such a methodology happens by means of open communication and minor adjustments to make things work as expected. In other words, methodology is driven by project teams. More importantly, a methodology that works for one distributed ecosystem may not work for another distributed ecosystem. This is because, for any methodology, while the basic tenets remain intact, the implementation details vary across ecosystems [1]. Hence, methodology is an internal affair of the project team.

This distributed Agile principle includes two dimensions: (1) a collaborative approach to tailoring [2] a methodology to fit a context, and (2) ongoing efforts to continuously improve the methodology in order to sustain throughput and quality.

2. Consistent usage of common tools improve productivity: Team members in a distributed team must have access to a standardized set of tools for engineering activities such as design, coding, static analysis, unit testing, build automation, test automation, defect tracking, and so on. Disparate tools result in compatibility issues and loss of information [1] and hence impact team productivity. Besides, they need to use such tools consistently in order to realize the benefits.

3. Infrastructure for communication and coordination is crucial: Team members of geographically distributed teams depend on phone calls, chat, email and video conferencing for communication. Also, they depend on web-based tools for agile project management, issue tracking, defect tracking, etc., It is crucial to have an infrastructure that supports distributed development in order to relieve teams from technical issues related to communication and coordination.

4. Knowledge management is the key to success: Assimilation, creation, dissemination, and regular upkeep of knowledge related to technology as well as domain elements of a project are critical to success in a distributed project. This is especially important in distributed agile projects. This is because teams have to focus consistently in delivering working software over short iterations at a sustainable pace, and respond to changes coming from business users at the same time. Systematic and consistent focus on knowledge management improves our ability to induct new joiners in order to expand teams and also to manage attrition effectively.

5. Quality is multi-dimensional, and owned by everybody: Quality can be seen in terms of intrinsic or internal quality and external quality. External quality is an attribute that relates to the end-user experience. It can be assessed and improved through black box testing and defect prevention. Internal quality is visible to various groups in the development team, such as designers, developers, maintainers and technical reviewers. Internal quality is invisible to end-users. It can be assessed and improved through reviews and

static analysis. Internal quality can be improved by defect prevention, as well as defect detection, followed by analysis and correction or defect fixing.

Quality can be improved from different dimensions or streams of activities, such as inspection of requirements, design reviews, functional testing, performance testing, security testing, compliance testing, exploratory testing etc. Agile teams understand this multi-dimensional aspect of quality and value the whole team approach. Obviously, the set of metrics or measures used to understand progress needs to be multi-dimensional. A single dimensional view of quality is simply not comprehensive enough. Also, in distributed teams team members from every location have to demonstrate a relentless focus on quality.

6. Distributed Agile requires an inclusive approach: Distributed Agile teams need to consider an inclusive approach in order to nurture distributed ecosystems. This is more important than the distribution of functional modules or user stories across sites. Facilitating a 'Base Camp'[2] [3] at a central location at the beginning of the project, and having an adequate travel budget for team members to travel across sites at regular intervals is the first step in ensuring inclusion.

One of the primary goals of facilitating a base camp is to create a shared vision [4] among project teams. Setting up the base camp involves forming a seed team with at least 1 team lead, one or two technical leads and a handful of engineers. Typically, members of this seed team are selected from distributed locations. They come together and spend 4 to 8 weeks depending on the size of the project, at a central location where the project initiates from. Setting up the base camp provides several benefits. On the project execution front this provides an opportunity to have adequate clarity on technical environment, tools and key engineering processes. On the people front this provides an opportunity for rapport building that can assure efficiency in resolution of issues and conflicts during the project.

Implementing distributed test drives or reviews, distributed retrospectives and distributed root cause analyses are also ways to nurture inclusion.

7. Governance is the backbone of successful distributed teams: Geographically distributed teams cannot function on their own. Collaborative governance is critical to the success of projects executed by distributed teams [6].

In case of projects executed across multiple geographic locations and time zones with employees of the project sponsor organization, external vendors and independent contractors, the complexity of governance increases multifold. Hence it is absolutely essential to form a governance team that comprises of representatives from all locations and works together as a single body in order to run distributed projects successfully. Governance has been one of the key success factors in distributed projects.

Even though every project needs well-defined milestones and goals, it is very critical to define success parameters at governance level. This helps distributed agile governance teams understand project success in terms of a common set of parameters.

Without this step, governance teams tend to focus on transactional issues and miss the big picture. While it is imperative to have a long-term view of the future, it is equally important to focus on early success. One way to accomplish this is to define success parameters beyond tested code and avoiding extensive low-value documentation which is subject to costly rework. To make this happen, distributed agile governance teams must have strong, visible commitment to the success of projects. Having a one-year roadmap and identifying milestones or events that can be measured against success every quarter is a way to ensure early success and mitigate risks.

8. Automation enables sustainable pace: Automation of engineering tasks such as build creation, test data creation, unit test execution, regression testing, test result analysis, etc., is necessary to optimize or avoid manual efforts spent on routine tasks. With automation, team members get adequate time to focus on critical tasks that need manual intervention. The significance of automation is greater in distributed teams as in collocated teams.

9. It is essential to streamline the accumulation and pay-off of technical debt: Distributed teams need to be aware, aligned and organized in managing technical debt in order to deliver maintainable, robust software [5]. [3]

Agile teams create business value by delivering working software at regular intervals. While doing so, they make design trade-offs in order to satisfy business reasons such as meeting a release schedule. Technical debt is the result of such decisions or trade-offs. When this happens, agile teams need to pay off the accumulated debt by improving designs during subsequent iterations in order to improve maintainability. This must happen in a systematic way so that technical debt does not swell up to cause damage to the project. To do this, geographically distributed teams need to establish a common understanding on what constitutes technical debt among all team members and maintain a prioritized list of tasks required to pay off technical debt. Also, it is very important to be proactive by considering practices such as architecture envisioning to avoid certain types of technical debt and architecture prototyping to identify technical risks and pay off technical debt early in the life cycle.

10. Iteration progression is a common phenomenon and ensuring early success is a collective responsibility: Distributed teams show progress over the first three or four iterations. Aiming for instantaneous results from the first iteration is an unrealistic. Ensuring early success is a collaborative responsibility of project as well as governance teams [6].

Conclusion

These ten principles presented in this short paper are the result of our experience over the past ten years. During the initial years, we perceived them as good practices and later found that these are essential for

every distributed agile project. As a result we arrived at these ten principles. In our industry, several large enterprises have started adopting agile in geographically distributed projects. This trend is going to continue over the next decade. Considering the ten principles discussed in this paper and adhering to them will certainly benefit geographically distributed teams.

References:

1. Rashina Hoda, Philippe Kruchten, James Noble, and Stuart Marshall, “Agility in Context”, Proceedings of OOPSLA '10, ACM, 2010.
2. Barry Boehm and Richard Turner, “Observations on Balancing Discipline and Agility”, Proceedings of the Agile Development Conference (ADC '03), IEEE Computer Society, 2003.
3. Raja Bavani, “Critical Success Factors in Distributed Agile for Outsourced Product Development”, Proceedings of the International Conference in Software Engineering (CONSEG-09), Computer Society of India, 2009.
4. Jim Highsmith, “What Is Agile Software Development?”, CrossTalk, October 2002.
5. Raja Bavani, “Distributed Agile, Agile Testing and Technical Debt”, IEEE Software, Vol. 29, No.6, November 2012.
6. Raja Bavani, “Governance of Distributed Agile Projects: 5 Steps to Ensure Early Success”, Agile Record, http://www.mindtree.com/sites/default/files/agilerecord_07_bavani.pdf July 2011, Issue 7.
7. Raja Bavani, “10 Principles for Success in Distributed Agile Delivery”, Cutter IT Journal, Vol. 26, No. 11, Nov 2013.