Change Management in IT Agile Projects

By

Tally Fruchtman Rossiter

Tally Fruchtman Rossiter, Progressive Insurance, March 9, 2019
ACMP Change Management 2019
April 28 - May 1, 2019, Orlando, Florida.

Introduction

IT Agile project delivery is changing the pace and structure of delivering technical functionality, which may call for a shift in the implementation of traditional Organizational Change Management.
This paper provides first-hand observations about these types of changes in IT project delivery practices and opportunities for practitioners in the Change Management field to positively embrace this shift.

IT Agile Software Development and Change Management

Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context (Agile Alliance, 2013).

In practice, Agile Software Development aims to increase project delivery speed and quality by providing customers with usable new features in smaller increments, delivered more frequently, with immediate feedback to ensure the functionality delivered meets their needs. Rather than planning for a *great big IT solution* that includes everything a customer wants (the "waterfall" method), the method is to "chunk out" smaller bits of functionality, prioritizing and delivering them at a regular cadence.

There are many methodologies with specific structures that can be selected to implement an Agile project (Sacolic, 2018).

In some ways, the Agile Software Development process has been compared with Edward Deming's Plan-Do-Check-Act (PDCA) process improvement cycle — deliver small, frequent pieces of functionality into production, check for solution effectiveness, and adjust work for the next cycle (Iterations, 2018; Hunter, 2008).

With an Agile approach, rather than defining all the deliverables and timelines in detail up front, the project defines high level goals and priorities, then focuses on delivery of "chunks" of functionality to meet those goals, allowing for customer feedback to change the specific details of the functionality delivered and the order by which it is delivered.

The shift to Agile projects follows a broader change in IT products and the need for their delivery to be faster and more responsive to customers and integrated system developments. Technologies rapidly evolving or not transparent to the user, such as infrastructure, can make it difficult to define the end product up front, and as a result make it difficult to understand the impact to users.

The Agile Transformation – From a Change Management Perspective

The traditional approach to Organizational Change Management in a project calls for an assessment of the stakeholders, impact, and risks up front. The project defines a change implementation plan aligned to the project delivery timeline and approach, and executes against that plan, making small adjustments to the plan based on feedback.

However, creating a well-defined change plan and executing against it in a linear fashion is becoming more difficult as the use of the Agile Software Development methodology becomes more common and the pace of IT project delivery increases.

Prosci© recently released an in-depth study of on Change Management practices with the transition *to* Agile projects and *in* Agile projects (Prosci, 2018; Creasy, 2018).

Some of the highlights listed for using Change Management *in* Agile projects include challenges in effectively adopting the Agile method to begin with, and the tendency to be less strategic about the people approach to adoption or operationalization.

A Change in Plan

A small database project with multiple downstream data users used up-front Sponsor, Stakeholder and Impact Analysis.

While speaking with stakeholders the project team discovered there were too many variables in the tool's capabilities and design to fully understand the impact prior to delivering some of the functionality. At the same time, a similar tool was introduced and sponsors changed.

Eventually the project was scaled back and closed without putting the intended capability into production.

Making the Shift

How Can Change Management Work with Agile Projects?

In her LinkedIn article (Gholami, 2018) Behnaz Gholami discusses whether "Change Management is Dead." Her conclusion is that organizations still need to support people undergoing change, and that perhaps the term "Change Management" should be adjusted.

The changes being seen in IT projects indicate that something has to shift. This does not mean that Change Management is no longer necessary, but it may indicate that the way it is done needs to change.

Changes happening with technology and project methodology may also offer some solutions for Change Management, specifically in these areas:

a. Agile structures providing supporting frameworks for Change Management activities
b. Change Management as a foundational skill
c. Automation

New Opportunities to Incorporate Change Management Activities

As part of the Agile project itself, some ceremonies and frameworks might provide a good fit with Organizational Change Management activities.

First, as part of Agile projects, the vision for the solution is defined at a high-level up front. Users and their needs are defined so that the project can prioritize high level groupings of functionality that would provide the most value, without going into detail around how each new feature would be delivered. This value conversation provides a good opportunity to gather a high level assessment of the level of change each feature will introduce.

Second, with the introduction of "standing" Agile teams, or teams that are fed new inventory of project backlog within a specific domain, the work of Organizational Change Management may become easier, as feedback from users is integrated into future releases of the product.

In the past, once an IT product was delivered to production, the user was left to adopt the changes to the process or technology. Sometimes an "implementation" business unit supported these efforts. If needed, a new project would be opened down the line to update or correct areas of concern.

In her article (Gholami, 2018) the author states that fundamental shifts must occur in terms of less management, and more support, of those who are undergoing change.

Continuous feedback channels for the Agile project team can promote this behavior, as well as a closer connection between users and IT projects.

In the same manner, operationalization, rather than being done at the end of a project, is something that happens as part of each sprint. Creating documentation and support must become part of the delivery of each feature. Some groups are using "hardening" sprints to prepare a product for production, or go through governance review. In the same way, Change Management can be done in smaller portions

for each feature. This adjustment will require a broader organizational (and customer) understanding of the variation in functionality provided for each release. Some products being delivered initially may not require broad Change Management support. At the same time, just like a feature may be released as a "beta" version, Change Management plans may be limited and not include every aspect of adoption when a product is first released.

Last, the cross training that occurs in Agile teams provides an excellent opportunity to share knowledge of Change Management among more team members. A developer that can write requirements also needs to understand what changes to the system may mean to the user, so they can write the requirement correctly, and potentially identify a solution for it.

A Foundational Skill

There are several models describing how a Change Management resource fits into a project team – either as part of the team, or consulting to the team (Prosci).

A Change Management resource within the project team is either defined as a separate role, or can be taken on by one of the project roles, most typically the Project Manager or the Business Analyst.

In an Agile project other project roles such as the tester, and developers need to have a much more in-depth understanding of the change that the end-user has to undergo with a new system in order to better define the work associated with each release of new functionality. If working in a SCRUM methodology, the Scrum Product Owner and Scrum Master take on some Change Management activities as well.

The following visuals portray the change in how project team members interact with stakeholders and end-users or teams representing end-users.

In traditional IT projects, the Change Management resource would interact with stakeholders outside the core project team, including the customers for the product delivered or those inside the company representing the customer. Change Management activities would be focused around the delivery and adoption of a completed product, after the majority of the development work has been completed. The development team often would have minimal interaction with the customer and other stakeholders.

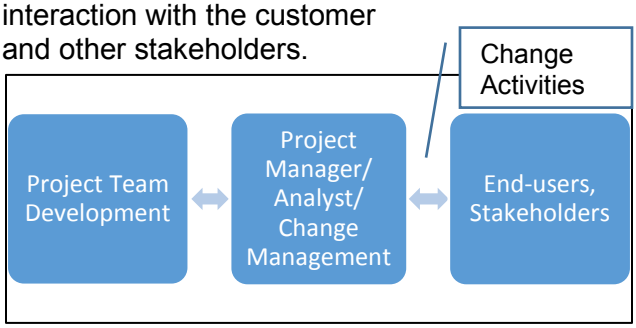

Figure 1 - Change Activities in Traditional IT Projects

Agile projects introduce a m[illegible] model, [illegible]hereby the custome[illegible]uct is either part of the [illegible]oject team, or works very closely with the project team, receiving more frequent updates. The product delivered may be modified multiple times during the life of the project, and the proje[illegible] team is all involved in reviewing and adjusting the product with the customer.

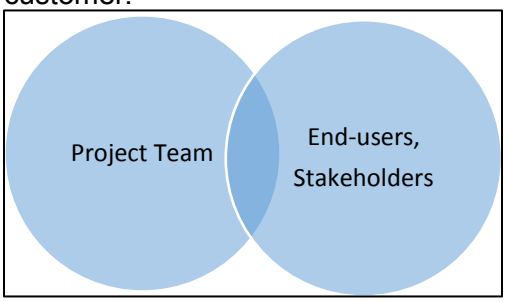

Figure 2 – Change Activities in Agile Projects

This means more individuals on the project team need to understand Change Management and how to plan for changes. Roles like the Scrum Product Owner and the developer need to help weigh the level of change that is being proposed as part of the "highest value feature" to correctly estimate the work that is to occur.

Automating Aspects of Change

In addition to the way an Agile project may support Change Management activities, there is a need to look at the speed and breadth of changes that technology will bring about and consider whether human involvement in delivery is always the best option, or where there are opportunities for automating Change Management activities. Those managing and receiving technology changes must find better ways to keep up with the pace of technological change.

Organizations are already looking to automation to address some elements of production-readiness that are standard such as security and performance testing. In the same manner, some aspects of Change Management could be automated into the product use itself. The tools themselves must become a mechanism for communicating change.

Rather than receiving an email every time a feature has been updated, users may see a notice of the change come up when they use the feature for the first time, similar to what mobile phone users experience today. Or, feedback may be automatically incorporated after each tool use, like ways that Skype and other online collaboration tools do it. This approach is much less disruptive to the user and ensures that the training and communication received is targeted only to the feature being used.

Automating communication and training into the product release cycle may offer a solution for organizations looking to maximize their people's time and attention, and ensure that manual Change Management activities can focus on the more difficult tasks of process and culture change.

Conclusion

In his post about adapting and adjusting Change Management in Agile (Creasy, 2018), Tim Creasy states that when supporting an initiative using Agile, the change practitioner must be more flexible and adaptable. He states, "There is a certain level of discomfort involved with letting go of 'perfect' Change Management strategies and plans. However, the rapid pace of a project using Agile means that Change Management practitioners must refine and focus their work — becoming more precise and efficient and knowing where to flex and where to relax the Change Management rigor."

As a team member in projects transitioning to Agile, one struggles with the discomfort of a looser change plan. However, the opportunity this shift presents for the practice to grow more flexible, and for more players to develop Change Management skills, makes the "What's-in-it-for-the-practice" an enticing proposition.

Balancing Risk, Planning, and Value

A traditional "Waterfall" project introduced LEAN and SCRUM concepts to increase speed.

As each new feature was introduced, a separate consideration was made for the feature's production readiness and adoption. The Change Management plan became similar to other deliverables - general at first, and more detailed around each deliverable, with feedback between deliverables.

Some deliverables required almost no "management", while some deliverables took longer because they required more focus on adoption.

References

Agile Alliance. (2013, June 8). *What is Agile Software Development?* Retrieved from Agile Alliance: https://www.agilealliance.org/agile101

Creasy, T. (2018). *Adapting and Adjusting Change Management in Agile.* Retrieved from Prosci: https://blog.prosci.com/adapting-and-adjusting-change-management-in-agile

Gholami, B. (2018, December 18). *Don't Twist it Please! One Simple Reason Why Change Management is in Question.* Retrieved from LinkedIn: https://www.linkedin.com/pulse/dont-twist-please-one-simple-reason-why-change-question-gholami

Hunter, J. (2008, September 16). *Agile PDSA.* Retrieved from Curious Cat: https://management.curiouscatblog.net/2008/09/16/agile-pdsa

Iterations. (2018, October). Retrieved from SAFe: https://www.scaledagileframework.com/iterations

Prosci. (2018). *Change Management Agile Report.* Retrieved from Prosci: https://store.prosci.com/change-management-agile-report.html

Prosci. (n.d.). *Change Management and Project Management Dimensions.* Retrieved from Prosci: https://www.prosci.com/resources/articles/change-management-and-project-management-dimensions

Sacolic, I. (2018, March 15). *What Is Agile Methodology Modern Software Development Explained.* Retrieved from InfoWorld: https://www.infoworld.com/article/3237508/agile-development/what-is-agile-methodology-modern-software-development-explained.html