Implementing Off-the-Shelf Solutions with an Agile Mindset

by Edith Trevino and William Sun

Agile methodologies originated as a way for software developers to improve the way they operate by implementing processes that focus on continuous improvement and delivering value to the user. However, for enterprise solutions, the majority of organizations don’t develop their own software; instead, opting to implement or customize off-the-shelf products already developed by a software firm.

Because off-the-shelf solution implementations mainly focus on gathering requirements for configuration instead of writing code to build the solution, we often see companies struggle to utilize the agile methodologies to full effect, resorting to a more traditional waterfall approach to design, testing, and deployment. From our past experiences, we have highlighted some key considerations to keep an organization agile during off-the-shelf implementations.

1. Establish Executive Buy-in and Empowerment

Although software implementations are typically IT-driven, derived process changes and outcomes impact multiple areas of an organization. Without executive buy-in, it is unlikely that you will obtain the expected results. As such, leaders should advocate on behalf of the initiative, ensure the necessary training, and allocate adequate resources in key roles to maximize success. While not everyone in the organization will work in agile teams, we advise that you establish a basic understanding of its principles and framework across the organization.
Creating successful, self-managed agile teams not only requires training, but also a cultural shift. Change management strategies should be put in place to maximize the success of this transition. We recommend designating an agile "champion" within the leadership team to ensure each member, particularly the product owner, feels empowered to make decisions and to step-in to remove major roadblocks along the process.

2. Build Clear Roles

Most companies that pursue an agile journey select the Scrum methodology as a framework to structure their teams. Each Scrum team is generally made up of three essential roles: Scrum Master, Product Owner and Project Team. Initial training should provide pertinent information relating to the responsibilities and expectations of each role. Some common mistakes we’ve seen companies make are: combining roles, assigning tasks to external resources that should implicitly be owned internally, or creating siloed Project Teams. Here are some high-level recommendations to consider before assigning roles:

3. Be Flexible with Agile Rules

When implementing greenfield off-the-shelf software, deploying functionality into production in a span of two or three-week sprints is not feasible. Design decisions which impact roles, processes, security and integration require numerous conversations which could take weeks or even months to finalize. “Done” in scrum terms will mean that a person or group is accountable for completing a specific task or design decision within a sprint, while delivering a corresponding piece of software to business users. This is certainly different from waterfall processes, where a hybrid waterfall approach developers or “production” copy because smaller pieces in a span of multiple sprints. The true value lies in the accountability that is created for all tasks and was received and applied in real-time.

If the level of project impact is significant, we recommend assigning a fourth role - an IT expert. The person assigned to this role should have deep knowledge of the company’s system architecture in order to deploy solution in smaller pieces in a span of multiple sprints. Don’t be too rigid - adopt a set of methodologies and best practices that enables you to successfully reach your objectives. Learn about your company’s capabilities and vision to ultimately understand your needs and provide better guidance towards your implementation strategy.

4. Manage A Project Without a Formal Plan
Agile methodologies emphasize the allocation of work based on priorities. Instead of a detailed, waterfall-style project plan, a backlog of stories is maintained, and the highest priority items are tackled each sprint. This allows the team to remain flexible and account for unanticipated user needs that arise during the project. When implementing an off-the-shelf solution, the project team should start with a standard, known backlog of features based on previous experience.

Operating without a formal project plan can feel uncomfortable. A lack of a formal plan can also make it more difficult to communicate project progress to others in an organization, many of whom may often be operating under a non-agile project. We recommend that project teams, based on previous experiences provided by a consulting partner, forecast high-level milestones and targets beforehand. However, it is also important to emphasize that these milestones are estimations instead of set-in-stone dates as in a normal waterfall-style project.

We have found that this hybrid approach to project planning works well and is more easily digested by the rest of the organization. Additionally, since implementations are not typically pure software development, it helps keep the project and team on track to an initial budget.

5. Provide Value in Smaller Pieces

With off-the-shelf solutions, it can be tempting to do a “big bang” style implementation, where every piece is designed beforehand and then released all at once. However, one of the key tenants of an agile project is to deliver value to the business as soon as possible. Breaking down functionality into smaller, value-add pieces makes this possible. Here are some common methods to effectively provide value sooner:

- **Release Functionality Modularly** – Release elements of a solution early and often. For instance, an agile team working on PLM might first release document management, then part and bill of material management, then supplier management, then compliance. Doing so will provide tangible value sooner, allow the project team to remain focused on one aspect of the system, and gather earlier feedback from actual system users to be incorporated into future designs.

- **Break Down Deployments** – Different regions or business units of an organization often have varying processes, regulatory concerns, and IT systems that need to be accommodated during a deployment. Rolling-out deployments in smaller releases can allow for more effective, tailored training and ensures that local site concerns can be addressed accordingly by the design. In addition, you can obtain learnings from early deployments and apply them to subsequent deployments.

- **Respond to Post Go-Live User Needs** – After a system is live, users obtain additional experience in the system, often clarifying needs and requesting functionality improvements. In a traditional, non-agile environment, these requests are often submitted to the project team then put on a release schedule that can be months away. With an agile mindset, the project team can implement high-value functionality requests more quickly, providing immediate impact to the business. Once the business users understand that post go-live changes can be implemented quickly, they are much more comfortable with starting with a non-perfect solution, which will shorten the initial design time.

6. Establish Clear Communication Channels with the Users
In an agile-style implementation, getting user feedback and responding quickly is key to success. With traditional waterfall projects, long lead times for new requests result in end-users feeling that their voices are not being heard. Establishing clear lines of communication is critical to being agile.

We recommend establishing a core agile team to make day-to-day project decisions and perform the actual project work, an extended team to represent other sites and business areas and provide feedback, and a steering committee to help guide the overall vision and break deadlocks. Additionally, a cadence should be established such that at the end of each agile sprint, a demonstration of the features worked on in the sprint is presented to the extended team. When implementing off-the-shelf software, releases can often be many months and sprints away from each other in contrast to software development where releases can often be deployed to production regularly (e.g., every two sprints). However, this should not stop the project team from demonstrating the sprint outputs for user feedback.

**Conclusion**

To keep pace with emerging technologies and trends, as well as budget constraints, leadership is increasingly requesting shorter implementation timelines, with the flexibility to modify requirements and scope. Rethinking foundational processes, structures and relationships is not a simple task, but, with the right guidance, it can mean a smooth transition to an agile organization. Based on our own observations, following these six points will help to make your transformational agile journey a successful one.

Originally published on January 8th, 2018

What's your view? Add your question or comment

Share
- Share on Facebook
- Share on Twitter
- Share on Google+
- Share in Email
- Download PDF

Topics: Adaptability, Agile, Flexibility, Implementation, Implementation Methodology, IP, IT, IT Implementation, PLM, SCRUM, Software Implementations, Strategy, Waterfall

**About the Authors**
Edith Trevino

Edith is a consultant with Kalypso, focused on helping clients implement innovation strategy and product lifecycle management (PLM).

More Viewpoints by Edith Trevino
William Sun

William is a Manager with Kalypso.
More Viewpoints by William Sun

Recommended for You

Related by Topic

Addressing Unique Change Management Challenges with PLM in Retail

Previously in Consumer Goods

Year in Review: Top 10 Innovation Viewpoints of 2017