JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

How Using Agile Can Help with Risk Management

ARTICLE

By Steve Berczuk | November 19, 2013

SUMMARY

Agile methods are one way to use iterations and frequent feedback to manage risk. Getting feedback early so that you can make corrections or change expectations isn't a new idea, but implementing a process that can give you both this feedback and the tools you need to make corrections is difficult for a number of reasons.

One reason you may be trying to adopt agile is that you are having trouble delivering software predictably in your business, technical, and team environment. One of the main differences between agile and traditional projects is their different approaches to managing and mitigating risk.

Software projects are complex and can quicky go off track. This isn't a new phenomenon. Consider that Fred Brooks in his 1975 book on software projects, *The Mythical Man Month,* asked, "How does a project get to be a year late? One day at a time."

Agile methods are one way to use iteration and frequent feedback to manage risk. Getting feedback early so that you can make corrections or change expectations isn't a new idea, but implementing a process that can give you both this feedback and the tools you need to make corrections is difficult for a number of reasons. Even though agile came to popular awareness in the late 1990s, it's still hard for teams to fully adopt agile. Part of the reason for this reluctance to adapt new ways of working is that the traditional approach to risk management is both deeply ingrained in the culture of many organizations and also appealing at some level.

Traditional Project Management

Non-agile projects attempt to manage risk by trying to anticipate problems, plan for them, and then move slowly to avoid the unexpected. Plan-driven approaches are focused on minimizing losses rather than maximizing gains, which according to some

We use cookies to ensure you get the best experience on our website.

Accept

Read More

We like to think that we can anticipate every contingency and deliver a plan that can effectively drive the project. Trying to work in the context of a plan-driven approach like this involves the following: doing a lot of up-front work, such as detailed requirements, design, and estimation; making decisions with risks to the plan as a primary consideration, rather than evolving business situations; and checking progress against the plan and considering whether to continue or cancel the project depending on the progress.

A plan-driven approach may be fine for very short-term projects when things are predictable. But things almost always change over time, and projects rarely go according to plan. Even something as simple and risk free as running an errand can be difficult to plan in a predictable fashion. With bigger projects we feel more of a need to plan things out because we feel that there is more at risk, and thus we should apply more control.

Some teams may choose to manage technical risk by scheduling more technologically risky work first, rather than more important features. This will mitigate technical risk, but will make it more likely that your project will suffer when there are changes to the business environment.

For example, you may decide to do a detailed database implementation to validate that your database design will support all of the features scheduled for the project. However, you may later discover that this work puts you behind schedule when you need to release, and as a result, you'll be missing essential business features.

Following a master plan is comforting, but it doesn't always work in the face of the various forces that a project deals with. There are technical challenges, team challenges, and changes in the business environment that make following a long-term master plan a risky approach, especially when you don't periodically re-evaluate your plan. Some of the things that can cause a plan to go awry include frameworks that don't work as you expect, staffing changes, a competitor introducing a new product, or a change in business rules.

Adherence to a plan can also present business risks. For example, you may be planning to introduce a new feature as part of your big semi-annual release when a competitor introduces a similar feature before you are finished. Unfortunately, you discover that the most important feature was planned to be completed last, so you can't deliver it in a timely manner.

Confidence in in a plan-driven approach leads you to build plans that are fragile, or at the very least, do not force you to develop robust, incremental plans. Quite often when the plan no longer seems to meet business needs, the only option you may have is to cancel a project, since the plan may be difficult to change.

Another risk with plan-driven approaches is that having a good plan that covers all the details as completely as you'd like takes a lot of time. This means implementation starts later, and if something causes you to change the plan once implementation starts you have wasted a lot of time.

When things go wrong when you are using a plan-driven approach you may hit a deadline with either a non-functional system or a system that works but doesn't implement the

We use cookies to ensure you get the best experience on our website.

I've seen many projects in which people worked late to implement a feature than ended up not being used. In some cases the team members works hard to finish as much as they could before a deadline only to deliver a system that wasn't successful because of a small feature that didn't get finished by the deadline.

Another common situation is that the team that delivers a somewhat complete system that was not tested thoroughly, so that it ends up being unusable. In the latter situation, a less complete, but better working system might have resulted in more business value. By focusing on managing risk and measuring the success of the project solely based on the goals of the original plan, you can lose opportunities to deliver useful systems that solve problems. With this in mind, people sometimes look to agile as a solution.

Agile Risk Management

The agile approach to risk management is based on frequent feedback and iteration. The development team and the product owners are communicating frequently in a controlled manner in order to help keep the project moving in a direction that delivers the highest value features first. Rather than defining all the steps in the plan, you spend time and energy defining the high-level goals and constraints of the project, and track progress against these goals and constraints.

You may still have a master plan that outlines the boundaries of what you're doing. The difference is that you revisit the plan periodically to monitor progress against that plan. You spend more time and energy planning the high-priority work that you're doing sooner.

Based on your progress and other business constraints you can then adjust your plan. In the example of running an errand, you might have time constraints and a few things that you really need to get at the store. If your child takes longer than you expected to get out of the house, you might then adjust your plan to only make the most important stops.

On an agile project you might revisit the plan every two weeks (or more frequently). To fully benefit from revising the plan periodically you also need to insist that you have a usable software system at the end of each iteration. The combination of frequently iterating on the plan and having continually working software means that you can deliver something sooner, even if your plan changes dramatically. Even when something goes wrong in an agile project, you have some faith that you can deliver a system that has implemented the highest priority features.

A challenge in adopting agile is to get everyone involved to agree that agile planning is a reasonable approach. Once you have done this, you need to change your project management and technical practices to make the approach work. This means establishing practices to allow for the iterative planning that is part of an agile project. An agile plan only works if the code is also nimble enough to adapt, so you need to establish the technical practices that make it possible to deploy a system frequently, and evolve code to meet changing requirements easily.

Team members can't be agile unless they acknowledge that there will always be uncertainties and that they cannot plan for everything. Rather than taking comfort in a plan, they take comfort in the knowledge that they can adapt to change and still be successful.

We use cookies to ensure you get the best experience on our website.

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

UPCOMING WEBINAR

LETS HANG!

FEATURED RESOURCES

AGILE

Implementing Agile eGuide

White Paper | StickyMinds

We use cookies to ensure you get the best experience on our website.

AGILE

Scrum Master Essentials eGuide

White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

0 comments

YOU MAY ALSO LIKE

AGILE

From Silence to Self-Organization: What Social Interaction Games Achieve That AI Cannot

Dinesh Sharma | June 17, 2026

AGILE

Why Your Agile Team Is Failing Even When Everyone Knows Their Stuff

Ella Mitkin | March 2, 2026

AGILE

Solving the Problem vs Treating the Symptoms

Payson Hall | January 16, 2026

Join the crew for FREE! Access exclusive content,

We use cookies to ensure you get the best experience on our website.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.