

Five Scary Things About Adopting Agile

by Mike Cohn • 26 Comments

Five Scary Things About Adopting Agile

I've never been a huge fan of horror movies. But I have always liked Halloween, especially as a kid. I loved dressing in a scary costume and trying to frighten my friends or other trick-or-treaters. And I admit to getting a bit of a thrill from walking through a well executed haunted house.

But, there are aspects to every agile transformation that can send a shiver down a spine--and those moments are not quite so thrilling. So let's take the fear factor away from five things that might make you want to run for cover along your agile journey--first by naming them and then by talking about why they aren't so scary once you understand how they work.

Eeek! Agile Has No Design Phase!

Architects and senior tech people are often frightened by the idea that there's no design phase in agile. While it's true that agile teams shun an upfront design phase, that does not mean that design doesn't occur.

Design on an agile project is characterized by two attributes: it is both *emergent* and *intentional*. An emergent design is one that comes into existence over time. Rather than being done all upfront in a single phase, design is an ongoing activity. The design emerges through the creation of the software.

But design is also intentional. This means the design does not emerge randomly. Design is guided by the intent of the senior technical people on a project or perhaps even someone in a designated architect role.

If the senior technical people are concerned about a particular part of the system, the product owner should prioritize a product backlog item or two in that area. This will allow the team to

explore that part of the system by building some small part of that system. Doing so will help identify the appropriate design in that area.

Allowing design to emerge guided by the intent of the team will reduce the terror of there being no design phase in agile.

Help! I'll Become a Generalist.

A common myth that has persisted since the earliest days of agile is that agile teams don't value specialists and want everyone to become a generalist. This is frightening to many people for two reasons: First, no one enjoys all types of work equally and second because of the concern it could impact future employability if someone can do four things adequately instead of one thing extremely well.

The idea that everyone becomes a generalist on an agile team is completely false. If I'm coaching a team that has the world's greatest JavaScript wizard, I'm going to want that superstar doing amazing things in JavaScript, not learning how to become a DBA.

Yes, agile teams value members who have multiple skills--the tester who can also write some JavaScript, for example. One of the easiest ways to balance the amount of work between two skills is to have someone who can do either type of work. If, for example, your team finds it hard to balance the amount of work in an iteration between programmers and testers, having someone who is adept at both coding and testing can solve that problem.

A goal in agile is the formation of cross-functional teams, which means the team has all skills needed to produce a finished, working deliverable within the iteration. Doing that does not require each person to have all the skills of a cross functional team.

If the thought of becoming a generalist makes your hair curl, understanding that a cross-functional team still allows for specialists should put your mind at ease.

Oh No! We Can't Plan or Predict Dates!

Management is often chilled to the bone by the idea that an agile team can't plan or predict

dates further ahead than the current iteration or sprint.

Fortunately, this does not need to be the case.

Yes, agile teams give up the false comfort of overly planned projects with intricately drawn Gantt charts showing a precise completion date way in the distant future. But that does not mean agile teams are unable to make plans or predict future delivery dates or scope.

A huge benefit of agile is that every iteration the team turns the crank on the entire development process. That is, they take an idea usually expressed as nothing more than a simple user story, and they fully implement that feature. This means that every few weeks, a team can measure its progress: They can know how much they got done.

Contrast this with a traditional development project, perhaps one with an analysis phase, a design phase, a coding phase and, finally, a testing phase. When that team measures its progress over a period of time, they are measuring only how fast they are at doing one (or perhaps two) types of work. How fast a team is at doing design says nothing of how fast the team will be at coding and testing.

The key with agile planning is embracing the uncertainty--admitting that it's impossible to know all the functionality that will be built before starting the project--and then adjusting for that in a variety of possible ways. When teams combine this realization with the ability to actually measure the amount of work done every iteration, it leads to reliable planning, which should calm the nerves of a management team daunted by the prospect of having no predictability.

Yikes! I Won't Have a Job!

The prospect of transitioning a team or company to agile is enough to scare the pants off some managers who are concerned they won't have a job after the transition is complete.

This is understandable. It would be dismaying to start a transition effort aimed at helping a company knowing that it may make your role superfluous.

However, I can clearly say that I have never helped a company adopt agile and then seen that

company say, “OK, we no longer need people with job title such-and-such. They’re all terminated.” It just doesn’t happen. Sure, a company may decide a specific job title is no longer needed or appropriate. But the individuals with those titles still have jobs. And in most cases, I believe they end up with better, more focused jobs. With that being said, it is true that some people will end up with less direct control over people or decisions and they may find that frustrating, even to the point of leaving the company.

But because I have never seen significant layoffs of any role or skill as part of transitioning to agile, I believe this fear is as misplaced as that of a zombie apocalypse.

Zoinks! Scrum Has Too Many Meetings!

Like many people, I really hate meetings. In fact, it’s largely why I do what I do. Most days I have no meetings at all. Pure bliss.

But even I can acknowledge that some meetings are important and useful. And that includes the four standard meetings of Scrum: the sprint planning meeting, the daily scrum, the review, and the retrospective.

The thought of all these meetings, however, can send some people into a cold sweat.

Here’s the problem with Scrum meetings--they have names. When something has a name, it’s easier to attack. In my experience, many teams who adopt Scrum had more meetings before Scrum. But the meetings didn’t have names and were mostly ad hoc meetings called to resolve specific issues.

Here’s an experiment I like and that you can easily do to see if you’re having significantly more meetings with Scrum. Randomly pick a month on your calendar from before you started adopting an agile approach. Open that month up in your calendar software and add up the time spent in meetings. Then add up the average time in your Scrum meetings and see how they compare.

Based on my experience, you might well be surprised by the results. But because those pre-agile meetings were not recurring, regular meetings, they didn’t have names and so they don’t

stick out in our memories the same way a named meeting, like “Sprint Planning,” does.

The key for overcoming the terrorizing thought of too much time spent in meetings is keeping the meeting short. Occasionally I’ll work with a team that I need to encourage to spend more time in a certain meeting. But most teams spend too much time in each of the standard meetings. Once they find ways to shorten them, those meetings should no longer frighten us out of our wits.

Relax...those Ghosts and Ghouls Aren’t Real.

While it can be fun to walk through a haunted house or dress in a scary costume, most of us can easily remember that it’s all make-believe. Those ghosts, ghouls, vampires, werewolves, and crazed lunatics with chainsaws aren’t real.

And neither are the five agile legends I’ve outlined here. And while no one can be faulted for being afraid at first, education and experience will pull the mask off the myths, leaving us reassured rather than unnerved.

What Have You Been Afraid Of?

What part of transitioning to agile did you find the most scary? How did you overcome your fear? Please share your thoughts in the comments section below.

Posted: October 31, 2017

Tagged: meetings, transitioning to agile, agile planning, design, job titles
