AGILE CONNECTION™
A TECHWELL COMMUNITY

Your online home for all things agile

Published on *Agile Connection* (http://www.agileconnection.com)

Summary: Jeffrey Payne sat down with Noel Wurst to discuss a range of topics, including advice for teams that are attempting agile for the first time, the importance of clear communication between teams, and the ways that security testing has changed alongside modern technology.

AddThis:

Security Testing in an Agile World: An Interview with Jeff Payne

By Noel Wurst - April 26, 2013

Noel : Do you think that teams that attempt an agile project for the first time fail to understand how difficult and intense agile can be? How can those teams better understand the breadth of what they're about to undertake?

Jeff : Yes. Teams often underestimate the amount of effort it will take. The first mistake many teams make is trying to "do agile," i.e., follow a prescriptive set of tasks and activities, instead of "becoming agile." Being agile is all about following the principles of agile, irrespective of the particular methods and tools that are used. This is one of the reasons why my Agile Fundamentals training course focuses a lot of attention on the principles of agile up front. There's nothing wrong with choosing a particular flavor of agile to begin your journey, but you will not be successful unless you tune and customize the approach you follow to your culture, people, market, and organizational structure over time.

The first step in understanding the breadth of what a team is about to undertake with agile is to understand that the entire organization must change for agile to be successful. This doesn't mean that a particular software development team cannot embrace agile while those around you do not, but that to get the full benefits of agile, everyone in the organization that touches the software must change at least some of their behaviors.

Noel: With communication between developers, testers, management, and the customer being so vital to the success of an agile project, what strategies would you recommend just in the area of communication, alone?

Jeff: First, I'm a big advocate of daily Scrum meetings. There are so many reasons why holding a brief standup every day is a good idea:

1. It forces everyone on the team to think about what they are doing on a regular basis and make incremental progress toward a goal

2. It drives issues, concerns, risks to the surface quickly so they can be addressed early before they become showstoppers

3. It holds team members accountable to each other which drives teams to meet their commitments and estimates

4. It's a very visible way for management to see the progress that is being made on a frequent basis

The key to an effective daily Scrum is to make sure each team member focuses on their *accomplishments* instead of just their activities. Use these meetings to track what was DONE, not worked on.

Second, use pairing to keep things fresh and transfer knowledge between team members. While pair programming is certainly one useful way of doing this, pair up a developer with a tester at the start of each new User Story. The developer can teach the tester a lot about what the design and code do, which helps the tester build better tests. The tester can teach the developer a lot about how to write good unit and acceptance tests so the code that's produced is of a higher quality.

Noel: One of your upcoming sessions is titled "Security Testing for Test Professionals." Everyone knows that security is important, but where are some areas concerning security that people, testers especially, may overlook, and therefore pose a threat to their project?

Jeff : Input validation. Many, many types of attacks leverage the fact that developers do not do an effective job of validating the integrity of the inputs their programs/interfaces accept. Too often input buffers can be overflowed, executable commands can be sent to a database, or inadequate authentication mechanisms are used to assure the person logging in is legitimate. Just cleaning up our inputs can go a long way toward making software more secure. These types of security issues are also ones that testers can understand and test for without understanding the details of the code.

Noel : Your bio states that you've actually testified before Congress on a number of security-related issues. What similarities do you see in regards to the need for increased security testing between the "simplest" software development project, and something as massive as the U.S. government?

Jeff : There is very simple software that is security critical. Likewise, there are massive U.S. government systems that aren't security critical at all. My point is that in security, we can't judge the need for security testing by the size of the organization or the software. We have to instead look at what the software does, what critical assets it has/protects, and what harm could come from its compromise. Based upon the outcome of such a risk assessment, a proper determination for the amount of security testing (and security assurance in general) can be made.

Noel : How has security testing changed over the years? In other words, are there concerns that someone who hasn't revisited their security testing requirements in the last couple of years, may be missing out on something that now needs to be paid attention?

Jeff : Often the biggest security issues in software come from the introduction of new technology. For instance, when the web emerged as a viable interface for applications, many legacy systems were "web enabled" to allow customers, administrators, support personnel, etc. to access and perform operations on legacy software through a web interface. Unfortunately these systems were never designed to be secure, as it was not expected that anyone would ever be able to access them remotely. This resulted in many, many legacy applications being compromised by malicious users.

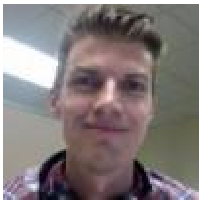
Today we are repeating this scenario with mobile devices. We are now build mobile applications and mobile clients that interact with our existing web-based applications. Mobile brings with it a whole new set of security issues and challenges that people are often not thinking about.

Jeff Payne is CEO and founder of Coveros, Inc., a software company that builds secure software applications using agile methods. Since its inception in 2008, Coveros has become a market leader in secure agile principles and has been recognized by Inc. magazine as one of the fastest growing private US companies. Prior to founding Coveros, Jeff was chairman of the board, CEO, and cofounder of Cigital, Inc., a market leader in software security consulting. Jeff has published more than thirty papers on software development and testing, and testified before Congress on issues of national importance, including intellectual property rights, cyberterrorism, and software quality.

Tags:
agile
agile transition
risk management
security
testing
Image:

AddThis:

About the author



Noel Wurst

A resident copywriter and editor for TechWell, SQE, and StickyMinds.com, Noel Wurst has written for numerous blogs, websites, newspapers, and magazines. Noel has presented educational conference sessions for those looking to become better writers. In his spare time, he can be found spending time with his wife and two sons—and tending to the food on his Big Green Egg. Noel eagerly looks forward to technology's future, while refusing to let go of the relics of the past.

View Profile