Reduce Manual Test Technical Debt

by Mike Cohn • 19 Comments

When a team that has relied mostly or entirely on manual testing decides to adopt Scrum, it will quickly discover how hard it is to run short sprints when there’s a lot of manual testing to be done each sprint. It will also realize that unless it does something drastic, the technical debt will continue to accumulate. Teams in this situation can follow a three-step process to extricate themselves from at least the worst of these problems:

1. Stop the bleeding
2. Stay current
3. Catch up

The first priority of a team with technical debt in the form of an over-reliance on manual testing is to stop the bleeding, stop things from getting worse. The best tourniquet is to find ways to automate some of what is being tested manually. To mix metaphors, teams should find the low-hanging fruit: tests that will be easy to automate but save a lot of manual effort. Brian Marick, a leading authority on testing and coauthor of the Agile Manifesto, has told me that “the real low-hanging fruit is often not automating some test execution but automating other testing tasks, like populating databases or automatic navigation to the page where you’ll start manual testing. You’re not reducing the number of manual tests, but you’re reducing the total time it takes to run them.”

After the bleeding has been stopped, the situation will no longer be getting worse from sprint to sprint. Manual tests still will be added each sprint but the team is finding enough low-hanging fruit each sprint to offset the time needed to run the new manual tests. At this point, it is time to move to step two: learning to stay current. During this phase, the team focuses on learning how to write and automate tests for whatever new features are added during the sprint. While doing this, no more debt is accumulating, so the situation isn’t getting any worse, but it’s not yet getting any better either. Learning to add automated tests in the same sprint as the feature will
be a new skill for the team. It won’t be as hard to learn as the initial skills were during the first phase but will require new discipline.

Eventually the team enters the final phase, which is when it catches up on additional outstanding testing debt. I generally tell teams that I coach that I don’t care how quickly the outstanding testing debt is brought down as long as it is indeed coming down. Obviously, I would prefer the debt to come down as fast as possible. But, by stating it this way I am emphasizing that what I am most concerned with are the first two steps: stop the bleeding and stay current. For more on testing with an agile process like Scrum, pick up a copy of *Succeeding with Agile*. 

---

Posted: December 21, 2009

**Tagged:** sprinting, programming, testing, agile manifesto