



Michael James is a software process mentor and Certified Scrum Trainer, focusing on the engineering practices that enable Agile project management. Having worked in the software industry for more than 20 years as a software developer (formerly “architect”), he has experience in automated testing that predates the Extreme Programming movement; formal, phased, high-ceremony processes based on DOD-STD-2167A; chaotic non-processes of the dot-com era; and Agile processes including Scrum and XP.

## Vampire Stories

Posted by Michael James on May 2, 2008 at 5:03 a.m.

A CSM course participant wrote to me about “vampire stories”: work that just can’t seem to be laid to rest.

*Dear Michael,*

*This course is really great which helps me clear many wrong concepts. I cannot wait to apply what I've learned on new projects.*

*We have been practising scrum for almost 2 year. But from my obersevation, it seems we are simply practising some activities of scrum such as dialy meeting, sprint startup meeting, sprint review, retrospectives without getting the correct output. For example, we have the experience many storoes cannot be closed by the end of the spring. The conclusion from the retrospectives is that we under-estimated the story(most likely a note from the developer who took most of the tasks of that story). But the next sprint, we are doing the same thing. We also have a lot of problem for software testing as well such as a lot of stories cannot be close because of no time to test.*

*So far, the concept of scrum seems quite clear to me now. But how to apply on a real project is quite challenge. Now, we just start a new project. I hope we can do better this time.*

*BTW, I have some questions:*

*Because of resource issue, I am playing multi-role: a scrum master, developer and leader of a team and software architect. There seems a lot of work to do for a scrum master.*

*1 as a scrum master, I need to work with PO to work out the sorted product backlog list(stories).*

*2 as a scrum master, I need to work with PO, testers to work out the acceptance tests.*

*But as a architect and leader of the team, there are many documentation work to do*

*1 requirements spec*

*2 design spec*

*For example, for a previous project, We worked out a very detailed design spec even with all screen layout designs during the project startup. But as the project moves on, there are a lot of changes later on. But the design spec is never got updated cause it seems we have not time to do that.*



Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

*So to me, it is quite blur. Do we still need a detailed design spec? From my experience, every one is expecting a good design spec cause it defines the interface between modules. But according to scrum, the design can change for each sprint. And hence updatting of such a document is really time-consuming. Especially, for a multi-role person.*

*Hence, I would like to know your advice on the documentation issue for a scrum project.*

Since I haven't seen your team working, I only have 50 percent certainty, but it appears some combination of these issues are occurring:

1. Poor collaboration on the team between the people writing code and the people testing code. This is the whole team's problem, and primarily a ScrumMaster responsibility to address.

Things that can help include working together in one team room, someone with strong "people skills" encouraging cooperation and pair programming (sometimes women are better at this than men), and improved Sprint Retrospectives (see Esther Derby's book *Agile Retrospectives*).

In our discussion, you mentioned a programmer was irritated by constant interruptions from a tester with questions about requirements. Wouldn't it be nice if writing the code is the only thing necessary to get products out the door? But as you know, you also need requirements analysis and testing. Testing is usually the bottleneck. People are sometimes surprised to discover they need more time for requirements analysis and testing than coding (which they see as the fun part). This is the whole team's responsibility.

Extremely Agile teams do the requirements analysis, testing, design, coding, integration, and review nearly simultaneously. These practices are described in Kent Beck's controversial books such as *Extreme Programming Explained: Embrace Change* (2nd Edition). I don't know if Extreme Programming (XP) will work for you, but I have seen it work better than anything else for other teams. In fact, Jeff Sutherland ("inventor" of Scrum) has observed he's never seen a hyper-productive Scrum team that wasn't also doing XP.



2. From your description and our conversation, I would agree in your case it's too much for you to take full responsibility of ScrumMaster, architect, and team lead at this time.
  
3. Your stories are probably too big. They're epics. It's helpful to keep the stories small in functional scope (but still a robust definition of "done" including testing and refactoring). You could try making new stories when you find alternate flows, exception paths, error cases, aesthetic quality issues, subjective UI experience issues, scalability, security, bells, whistles, etc.

It's hard for me to judge the amount of design documentation you will need. Agile teams tend to have a smaller amount of more relevant documentation. Extreme Programming teams use executable tests as functional specs, which is just brilliant once you develop the skills and habits. I would put this question to your team during a Sprint Retrospective.

--mj

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

Copyright © 2008 Danube Technologies, Inc.

danube.com