

This handout discusses a few common Scrum dysfunctions that are evident in teams that are just starting to use Scrum.

Change Scrum Too Early

One general piece of advice I have for teams practicing Scrum is to follow the rules, roles, principles and other constructs for some time and see how it applies to your environment. You will see the benefits soon. This is where you begin to understand the Whys of Scrum. An example is: what have we gained from retrospectives? What would happen if we did not have retrospectives? What if we have the daily Scrum on a weekly basis?

Scrum events are mandatory, and, if you practice Scrum, you will notice why you need those events. The timeboxes in Scrum events cannot be violated. And the Scrum roles are required.

Hardening Sprint

Many Scrum teams fall into the trap of deferring testing activities, performance testing, and other code quality enhancements to a future sprint called the “hardening” sprint. That is an antipattern. If you code in one Sprint and test in another, you are doing waterfall. First, you run the risk of being away from your stakeholders for an extended time. Second, you delay the delivery of business value and as a result run the risk of ignoring change in market conditions and requirements and losing to competition. Third, you have the problem of reduced visibility because your progress is not well understood. And you could have more problems than you think when you enter the “hardening” sprint. The general advice is to build and test in the same sprint and deliver business value at the end of each sprint. If you are implementing technical items like upgrading a class library version, mix those with items with business value in each sprint.

Sprint Zero

Teams must do some preparatory work at the beginning of any product development effort. Common examples of such work include: defining architecture standards, key architectural mechanisms, building coding standards, and more. Many teams do this in what is called Sprint Zero. Scrum purists do not like that term. There is no Sprint Zero in Scrum. So, what do you need to get started? A high-level product vision and a team are necessary. When do we execute this preparatory work? It needs to be done with other items of business value. That way, you can ensure that there is some business value delivered at the end of each sprint.

Defer Testing

I worked with a team where the QA activities were always being executed towards the end of the sprint and the issues were being fixed on the weekends. This was obviously causing stress and frustration because the testers had to work long hours, and on the weekends. The general advice is to test your code as you develop it. In fact, you should have your product owner in feature preview activities

throughout the sprint. Engineering practices such as automated testing and test-driven development will help in parallelizing QA activities with development. And if you have T-shaped people with coding and testing skills, developing and testing in parallel will be easier.

Scrum Master Is Project Manager

Is there a project manager role in Scrum? No. Can you have a project manager in an organization implementing Scrum? Yes, absolutely. Project managers are not part of the Scrum team. A project manager with servant leadership qualities is a great asset to any organization.

A Scrum master has a lot of responsibilities and the same is applicable to a project manager. We already covered the responsibilities of a Scrum master. A project manager may have responsibilities at a higher level and may have different sets of responsibilities, such as overall schedule management, contract negotiation, dealing with legal issues, and budgeting.

The two roles are different and the same person acting as both is a dysfunction in most cases. If you work for a company with three persons, you probably won't like the idea of hiring two persons for two different roles, but in most other cases, I don't see a reason to mix the two roles and make one person both Scrum master and project manager. It's too much work. Also, if you are in a control-and-command type of project management role, you will be a Dr. Jekyll as a PM and a Mr. Hyde as a Scrum master because you will direct people at one moment and be a servant leader at another. In general, fulfilling both responsibilities will be tricky and overwhelming and is not recommended. Also, if you are a project manager, it is a good idea to be an agile PM and be a servant leader.

Long Sprints

The last one I'd like to cover is the idea of sprints longer than 30 days. This is the same as waterfall and has the same type of issues as the waterfall model. Reduced visibility, increased risk of being away from stakeholders, losing to competition, and loss of business value due to changed requirements are many of the reasons you should keep your sprint length less than 30 calendar days. In fact, I recommend a sprint length of two weeks or less.