Chapter 10. **Implementing The Process**

How to add quality to your process?

In this chapter, I document ideas for integrating the activities of the process into an existing development effort.

I want to make it clear that there is no suggestion that a development effort should adopt every activity described in this book. Each activity is stand-alone work item with no dependencies other activities outside of the Scrum framework. Each activity encourages different quality improvements that are proportional to the extent to which it is implemented and the need for process improvement in that area.

- **I have never worked on an agile development project where every activity described in this book was adopted.**

None of these activities is free, and the expense may turn out to be more than the benefit gained due to the improvement in quality. The rewards for improving quality are different for every market. A highly regulated market may insist on quality before the product can be released. A brand new venture may include no explicit quality activities at first, but as the customer base increases, you may want to gradually improve quality over time.

Program management should determine how much to invest in quality while the project team can identify which areas are going maximize the return on this investment.

Using the best practices of incremental development, implementation of Quality with Agile activities should be introduced in small chunks. Before trying to improve quality, I recommend that your agile process (such as Scrum) is implemented and delivering incremental software builds. Only after you have a working process should you add quality improvement activities.

Continuous improvement should be an agenda item during the sprint retrospective meeting. This meeting is where the status of implementing the process is discussed. Suggestions for process improvement are captured from this meeting. Process improvement user stories are created and tracked in a process backlog (this may be the product backlog).

Create a process backlog (or create a template for a process type user story in the product backlog) and populate it with an epic for each Quality with Agile activity. The epic captures the activity (as described in section 7.4). When budget is approved, the epic is assigned to someone who will be responsible for the adoption of that activity. That person breaks the epic into user stories for:

- Understanding the activity impact to the current process.
- Identifying what we do today that would be improved by this activity.
- Investigating what the benefit is to adopt the activity and is it worth the cost to implement.
- Performing a gap analysis between the current and proposed process.
- Steps to implement the process.
- Measuring the impact to the process as a result of adopting the activity.

The above list is not complete. This list is a suggestion for a sequence of steps that might be used to adopt activities the Quality with Agile process.

At the completion of each user story, an approval is required to move to the next user story in the sequence of steps captured by the epic.

An example process adoption backlog may be similar to that shown in Figure 71.

Process Adoption Backlog			
Epic	**State**	**User Story**	**State**
Groom Backlog	Approved	Understand the activity	In progress
		Determine Improvements	New
		Investigate Benefit	New
		Perform Gap Analysis	New
		Recommend Steps For Implementation	New
		Measure Impact Of Adopting The Activity	New
Elicit Business Needs	New		
Design Architecture	New		
Define User Experience	New		
Maintain Requirements	New		
Test Build	New		
Deploy Build	New		
Document Release	Complete	Understand the activity	Done
		Determine Improvements	Done
		Investigate Benefit	Done
		Perform Gap Analysis	Done
		Implement Activity	Done
		Measure Activity Adoption	Done

Figure 71 - Process Adoption Backlog

Each Quality with Agile activity is captured by an epic. Approved epics have child user stories that capture the steps to implement the activity. Approved epics are listed according to their priority from the top of the backlog.

- **Although the completed epic is shown at the bottom of the backlog, completed epics are often removed from the backlog.**

The process adoption backlog may be integrated with the product backlog and user stories are pulled into Scrum development sprints. By integrating the process activity backlog with the product backlog, the steps to adopt the process must be captured as sprint sized user stories.

Prioritize epics in the order that the team wants to implement the process activities. Priority is ordered by the return on investment of adopting the process. This return on investment is dependent on issues with your current development environment and skillsets that can fill the appropriate roles.

For example, you may already have a person on the project that is experienced with data modeling tools. That person may want to take responsibility for implementation of the Maintain Requirements activity.

An example epic for the Maintain Requirements activity might be that shown in Figure 72.

Adopt Maintain Requirements	
For	business analyst
who wants	to maintain a logical view of the product
so that	we know the data that is used by the product and the dependencies between that data.
unlike	having to execute or search the software in order to find data and its dependencies
Additional information	
Tester	I need to know the data dependencies so that I am able to fully test the impacts to the system as a result of changing data.

Figure 72 – Example Maintain Requirements Epic

The steps for implementing the activity described by the epic are captured with user stories. Each user story delivers something tangible to the current development process. For example:

1. Create a logical view that captures the data used by the product

- **Note that it is not necessary to create model views in any particular order. The logical view provides benefit without a business, system or deployment view in the model.**

2. Create a class diagram that shows the objects manipulated by the product and the relationships between those objects.

- **If the work needed to create class diagrams for the whole product is estimated to take longer than 1 sprint then you could break the product into prioritized subsystems.**

3. Populate the classes in the logical view with data according to the priority of system use cases that are in development

- **The detail that is included with the data can be broken into user stories.**

Implementation of process should follow the same cycle as the development sprint. Determine the resources that you want spend on process user stories during the next development sprint planning meeting. If the resource is a member of the development team, you should pull the user story into the development sprint during the sprint planning meeting. If the resource is not a member of the development team, then create a sprint and pull the appropriate user stories into that sprint. The process sprint should follow the same cycle as the development sprint, such that progress of process adoption can be discussed during the Scrum retrospective meeting. The process sprint includes all the same ceremonies as the development sprint. Since the same people are present, it makes sense to combine these ceremonies (sprint review, daily standup, sprint retrospective and sprint planning) for both the process sprint and the development sprint.

- **An alternative may be to use a Kanban lifecycle for process user story adoption.**

In this manner, as the Quality with Agile activities are introduced, there should be no change to the work performed by the development team each sprint. The same metrics can be used to measure burn-down and collect data for other reports.

What should change is the amount of time it takes to produce a defect free release. Development becomes more efficient because the time spent on doing rework is reduced. Measure changes in productivity and monitor feedback from the customer in order to determine if the changes to the process are working.