MONTHLY NEWSLETTER

April 2008 - Pragmatic Software Newsletters

Agile Scrum - Understanding Scrum Rules

Many of us have experienced projects that drag on much longer than expected and cost more than planned. Companies looking to improve their software development processes are now exploring how Agile can help their Enterprise more reliably deliver software quickly, iteratively and with a feature set that hits that mark. While Agile has different "flavors", Scrum is one process for implementing Agile. This newsletter is one in a series of newsletters that will discuss the Agile Scrum process and will end with variants of Scrum that can be used to aid in improving your software releases. Here are the prior newsletters in this series:

- **Feb 2008: Agile Scrum - An Overview**
  http://www.pragmaticsw.com/newsletters/Newsletter_2008_02_SP.htm
- **Mar 2008: Agile Scrum - Team Composition**
  http://www.pragmaticsw.com/newsletters/Newsletter_2008_03_SP.htm

Understanding Scrum Rules

For Agile Scrum to be successful, you must have buy-in from the highest stake holders to the individuals doing the work and every individual must follow the rules. Below are the rules for Scrum:

- **Obtain Number of Hours Commitment up Front** - Before beginning an Agile 30-day sprint, each team member must commit to a certain number of hours for the 30 day sprint.
- **Gather Requirements / Estimates up Front** - The Product Manager will specify the sprint goal and the team will gather the requirements and provide an estimate up front. Once the estimates are done, the requirements are prioritized and only the ones that will fit into the sprint are worked on (based on estimated hours of all tasks vs. hours committed to by team members).
- **Enter Time Daily** - Each person on the team agrees to enter their actual hours and estimated hours remaining EVERY DAY.
- **Daily Builds** - Each programmer will check code in daily or more frequently if possible. The checked in code must be compilable. An automated process will create daily builds to prevent manual merging of code and allowing the Quality Assurance Engineer to test features of the sprint.
- **No new Requirements for a Sprint** - No new requirements can enter into the sprint unless all features of the sprint are completed. Management and other parties that are not directly involved in completing the features for the sprint are not allowed to direct actions of team members on items not included in the sprint. If an emergency feature is required by management, the sprint must be aborted and a new sprint must begin with the new feature set.
- **Keep the Daily Scrum Meetings Short** - Daily Scrum meetings are held to determine what things were done since the last Scrum meeting, what things will be done in the next Scrum meeting and what impediments stand in the way of any person on the team. The Daily Scrum meeting is designed to be completed in 15 minutes. If it takes 30 minutes, this is OK, but it should not extend longer than that without a solid business reason for it. Team members that show up to Daily Scrum meetings late are required to pay the Scrum Master a $1 fine.

- **Code Inspections are Paramount** - Upon completing a particular feature, the programmer should illustrate the feature to the team and show the code that drives the feature. The team should inspect the code for re-usability, cleanness, and adherence to established coding standards.

What's Next?

Upcoming newsletters will discuss the following topics:

- Agile Scrum - Scrum Kickoff and Product Backlog
- Agile Scrum - The Daily Scrum Meeting
- Agile Scrum - Reporting and Metrics
- Agile Scrum - Retrospectives
- Agile Scrum - Site specific variants of Scrum

Helpful Templates

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Pragmatic Agile Development - http://www.pragmaticsw.com/PADOverview.pdf**
- **Software Development /QA Templates - http://www.pragmaticsw.com/Templates.asp**
- **Software Planner - http://www.SoftwarePlanner.com**
- **Defect Tracker - http://www.DefectTracker.com**

About the Author

Steve Miller is the President of Pragmatic Software (http://www.PragmaticSW.com). With over 23 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.asp. Steve's email is steve.miller@PragmaticSW.com.