

A Roadmap for Using Agile Development in a Traditional Environment

Barbara Streiffer^{} and Thomas Starbird[†]*

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

Sven Grenander[‡]

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

One of the newer classes of software engineering techniques is called “Agile Development”. In Agile Development software engineers take small implementation steps and, in some cases, they program in pairs. In addition, they develop automatic tests prior to implementing their small functional piece. Agile Development focuses on rapid turnaround, incremental planning, customer involvement and continuous integration. Agile Development is not the traditional waterfall method or even a rapid prototyping method (although this methodology is closer to Agile Development). At Jet Propulsion Laboratory (JPL) a few groups have begun Agile Development software implementations. The difficulty with this approach becomes apparent when Agile Development is used in an organization that has specific criteria and requirements handed down for how software development is to be performed. The work at the JPL is performed for the National Aeronautics and Space Agency (NASA). Both organizations have specific requirements, rules and processes for developing software. This paper will discuss the some of the initial uses of the Agile Development methodology, the spread of this method and the current status of the successful incorporation into the current JPL development policies.

I. Introduction

Agile Development is a new software methodology that is gaining acceptance throughout the software community. Most Agile developments are used to reduce risk and to produce usable software in short time increments. Generally, the time span lasts one to four weeks. During that time a full life cycle is performed on a small set of functional capabilities. The goal is to have working software at the end of each cycle. The customer is a key part of the process and participates in all aspects of the development process. Agile Development is often considered the antithesis of the Waterfall Methodology. In the Waterfall approach requirements analysis, design, implementation and testing are performed in a linear fashion and only once. Usually the customer only participates in the requirements and design phases and then again during performance testing. The major difference is that the Waterfall approach is a more predictive approach in being able to determine schedule, deliveries and in general all the functional capabilities and the Agile approach is more adaptive and deals better with changes in requirements and with user feedback.

II. Agile Development at JPL

The beginnings of the one of JPL’s agile developments started in two research and development tasks sponsored by the Mars Technology Program in 2004. These tasks were to look into ways of infusing technology into the Mars Science Laboratory (MSL) ground system software. MSL is the next JPL rover going to Mars in 2009. One of these two tasks, called Ensemble, uses the Agile Development approach. Even though the other group, called the Next Generation Uplink Planning System (NGUPS), hasn’t followed many of the precepts of Agile Development, they have incorporated the work from the Ensemble task and have refactored legacy tools to be used with the software from the Ensemble task.

^{*} Software System Engineer, Section 317, M/S 301-250 D 4800 Oak Grove Dr., Pasadena, CA 91109 USA.

[†] Principal Investigator, Section 317, M/S 301-250 D 4800 Oak Grove Dr., Pasadena, CA 91109 USA.

[‡] System Engineer, Section 317, M/S 301-250 D 4800 Oak Grove Dr., Pasadena, CA 91109 USA.

Typically, the Ensemble team meets twice a week. One of the meetings is a status meeting where the preceding week's work is discussed and the next week's work is planned. The other meeting is a design meeting to discuss various aspects of the work that is currently being implemented. The team often works in pairs. The Ensemble team communicates with other members of the development team and with customers and tries to document its work in Confluence. Confluence, a wiki, is server software that is used to create and modify web page content. In addition issues are tracked in Jira, issue-tracking software. One aspect of Agile Development is that it produces less documentation than other methods because it relies on face-to-face communication. The Ensemble team addresses this criticism using by Confluence and JIRA. The NGUPS team is responsible for integrating the Ensemble work with legacy tools. The legacy tools have been altered to work effectively with the Ensemble as well as provide interprocess communication among all the tools. Periodically, members from both teams participate in pair programming depending on the "story" (functional element) to be implemented.

As the work from these two groups began to be noticed for its excellence, a third group decided to adopt the work and further its development so that it could become multi-mission ground sequencing software for in-situ missions such as rovers. Until now the multi-mission ground system software has been developed according to all the rules and precepts of a traditional system including the traditional set of documents, reviews, and other gates that are not typically part of the Agile Development process – at least not part of it in a traditional sense.

III. Traditional Approach

All NASA programs must follow NASA Procedural Requirements (NPRs). These requirements are for all development phases of projects. For software projects NASA developed NPR 7150.2. California Institute of Technology's (Caltech's) JPL has developed a response to NPR 7150.2 called the Software Development Requirements (SDR). The SDR has requirements that correspond to the applicable NASA requirements in NPR 7150.2. All software development at JPL is required to be in compliance with the SDR. The SDR is also compliant with the goals of Capability Maturity Model Integrated (CMMI). JPL didn't have a process that mapped Agile Development to the SDR.

At this point in time when JPL is working on being CMMI compliant, there is a support group called the Software Quality Improvement group. This group is to help each software development team to be compliant with the SDR and CMMI. This group has been trained in software development processes, the SDR, NASA NPR's and CMMI. Their goal is to work with each project to develop a series of processes including documentation that works.

IV. Combining the Two Approaches

The task has become one of how to marry the two techniques. Currently, there have been meetings and efforts to understand this new development model. The SQI group held a series of weekly meetings to understand how the Ensemble group has implemented the Agile Development methodology. The System Engineers from the NGUPS team and the Ensemble team have started work on documenting the implementation of Agile Development in a Confluence document called the Ensemble Software Development Process. In the newly created document the basic tenants of Agile Development have been stated with some modifications. Input from the SQI group has been received and accommodated. In general, because the Ensemble work is part of a larger system there are additional requirements on the development technique that must be met in the areas of planning, reviews, testing, etc.

For example, the process discusses planning of the software. It does talk about planning in weekly increments. However, it also deals with high level planning that creates a long-term agreement on release dates and general contents of the software at a high level functional capability. The next level of planning covers the planning effort that takes place after a delivery or a demonstration of the software. At this level of planning the detailed functional requirements are defined with the aid and review of the customer. These requirements are planned to be documented in JIRA. Additionally the division between core functions (project-independent ones) and project-specific functions is drawn. The final level of planning is the normal weekly Agile Development planning that occurs as part of the Agile Development Process.

In the area of reviews the Ensemble implementation of Agile Development calls for weekly status reviews that compare plans versus accomplishments. These meetings use metrics of schedule deviance, team velocity and budget variance to determine progress. The quality metrics of determining the number/kind of software anomalies and the number/kind of requirements implemented are also assessed. The JIRA issue tracking system allows these metrics to be accumulated. In addition to these reviews the group also participates in monthly manager reviews and typically holds demonstrations twice a year for stakeholders.

The Ensemble group performs the traditional life cycle for development. It includes requirements development, design both high level and detail level, implementation, test and delivery. However, these steps are performed weekly instead of once for each stage.

Testing and delivery is handled in stages as well. For any level of delivery automatic tests are run as regression testing. In addition, prior to delivery every developer is responsible for developing unit tests and running them. There are four stages in the delivery portion of the process. There is the stage where the developers deliver to an area that they control. In this stage software can be delivered as frequently as the developers have finished the implementation and have tested the new software. It is delivered to a loosely controlled configuration managed system. The next stage is a more tightly controlled delivery area. This area is where the software is integrated together and is used for preliminary acceptance testing. At some point in time the software then advances to a more tightly controlled area for formal acceptance testing. Finally it is delivered to the project area for user acceptance testing. This delivery scheme is typical of many of JPL software projects and has not been significantly changed from the Waterfall methodology. In JPL legacy systems it is possible to test the core aspects of the software separately from the project specific pieces because the core software is written in code and the adaptation is written in an interpretive language. However, since most of the adaptation work for new software is now in terms of actual code rather than being written in an interpretive language, it is often not possible to test the two pieces separately.

V. Conclusion

Currently there are a few relatively small areas that still need to be worked in having the Ensemble Software Development Process be totally compliant with the SDR. However, significant progress has been made in that direction and it is expected that the Ensemble Development Process will be totally compliant. A path has been created and the groups have been working their way to a resolution. Agile Development has become a reality in a traditional system (with a few compromises).