Lean Mission Operations Systems Design - Using Agile and Lean Development Principles for Mission Operations Design and Development

Jay Trimble
NASA Ames Research Center, Mountain View, CA, 94035, USA

The Resource Prospector Mission seeks to rove the lunar surface with an in-situ resource utilization payload in search of volatiles at a polar region. The mission operations system (MOS) will need to perform the short-duration mission while taking advantage of the near real time control that the short one-way light time to the Moon provides. To maximize our use of limited resources for the design and development of the MOS we are utilizing agile and lean methods derived from our previous experience with applying these methods to software. By using methods such as "say it then sim it" we will spend less time in meetings and more time focused on the one outcome that counts – the effective utilization of our assets on the Moon to meet mission objectives.

I. Introduction

THE Resource Prospector Mission (RPM), which passed its Mission Concept Review in the Fall of 2013, is a NASA Advanced Exploration Systems (AES) Mission to fly a rover and an in-situ resource utilization (ISRU) payload to the lunar surface in 2019. The mission is a collaboration of multiple NASA centers and external partners. At NASA Ames Research Center, we, in collaboration with our partners, are applying lessons learned from agile and lean software development to the design and development of the mission operations system (MOS) and the ground data system (GDS).

NASA MOS and GDS design practice utilizes a waterfall model for design and implementation. Requirements are specified, systems are designed and documented, then they are built. Designs are validated through a series of reviews, including, but not limited to, preliminary design review (PDR), critical design review (CDR), and an operations readiness review (ORR). CDR marks the transition from design to implementation. An MOS must be ready for the start of training. While changes are possible after that time, they are limited in scope. The system designed after CDR is very close to the system that will be deployed for the mission. The design is effectively frozen and configuration controlled before the start of simulations and training. However, the design work that went into the configuration-controlled version is based on requirements largely derived from meetings. We learn more in simulation and training about how we want to operate than we do in all of the meetings, but we have little capability for change at that point. So, while the traditional methods have proven highly effective in enabling safe and productive mission operations, modern methods have the potential to create more effective mission operations systems at lower cost.

For RPM we are incorporating agile and lean methods proven in software design and development to the whole MOS, including the design of the team, interfaces, operational decisions and calls, procedures, and mission processes. Based on our software experience, we expect these methods to result in a more effective mission operations system with a lower development cost than traditional methods, while preserving the benefits of a structured, safe, and repeatable process.

II. Agile and Lean

Before going further, we need to briefly discuss the relationship between agile and lean. Agile methods originated in the software world, where they were a reaction against heavyweight processes that attempted to control the software process, but often resulted in a product that was not what the customer wanted or needed. By contrast, lean comes from Lean Manufacturing and its associated principles, used to eliminate waste and improve quality in manufacturing. Lean UX applies lean principles to the development of user experience for software and product design. For more information on agile and lean, see [1,2,3]. People will ask the question, "Should we use agile or lean?"

In fact, agile and lean processes work harmoniously together and are not an either/or choice. The team should tailor and choose the technical and management processes that work best for their goals. We will briefly summarize our team's experience with user-centered agile software development, the principles we derived and practiced, then discuss lean principles and their application to RPM MOS design.

A. Agile Methods

Agile has gained significant traction in software development and is now common practice in many industries. Agile principles are well documented [2]. In practice, agile methods must be tailored to the context of the work and the environment. Our group at NASA Ames tailored agile principles and integrated our own participatory design practice to support the development of Mission Control Technologies (MCT) software [3,4]. These principles became the basis of our software development work.

The basic principles of agile, as documented in the agile manifest, are a good starting point:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Over a period of two years[3] we tailored these principles to our environment – the design, development, and delivery of a cross-platform software framework for spacecraft operations. Here are the key principles:

- The measure of progress is working code (not reviews, not slide decks)
- Demonstrations, not presentations
- Constant engagement with the customer
    - o Nightly builds
    - o Three-week delivery cycle, twelve-week release cycle
    - o Direct customer-to-design/development team interaction, triggered by nightly build and delivery cycles
- Constant prioritization within the bounds of the product road map
- Regular deliveries – the train must leave on time
- Rate of progress is demonstrated to the customer through constantly exposing the working code, rather than complicated metrics on slides
- Software must go through all mission certification processes prior to deployment in an operational environment

The agile processes outlined above improved our software quality, lowered our costs, increased the satisfaction of the team members involved, and created a closer relationship with our immediate customers. The customer was involved at multiple levels. With the nightly build, the customer could see our progress every day. Equally important, they were able to give daily feedback on feature development, allowing us to modify features to their satisfaction before delivery. Formal feature acceptance occurred with each three-week deliverable. If the customer did not accept a feature, it was further developed in the next iteration. The twelve-week cycle constituted a release. The final three-week iteration in a release focused on quality and bug fixes over new features. The continuing customer engagement led to ongoing feature prioritization, within the bounds of our agreed-upon road map.

We found that it was vital to be consistent, hence we did not delay a shipment – the only exception was if the software did not work. For features that were not ready, we removed them and included them when they were ready. The train must leave on time, and it leaves often enough that it's easy to get on the next train – always ship on time, ship often – if a feature is not ready, include it on the next shipment or when it's ready.

Figure 1 compares the agile process we used with a traditional process. Note that nothing in agile precludes having reviews. Reviews can still serve as discrete check points in the context of the continuous agile process. Also note the many constant quality drivers in agile – continuous customer engagement, short cycle times, a twenty-four-hour test before each delivery, two "hackathons" in each three-week cycle. A hackathon in our process was where the whole team used the product at the same time to test scalability and performance. It functioned as a high-stress test for the software (and sometimes for the team).

These development methods and principles, proven for the GDS, may be adapted for the MOS processes and procedures.

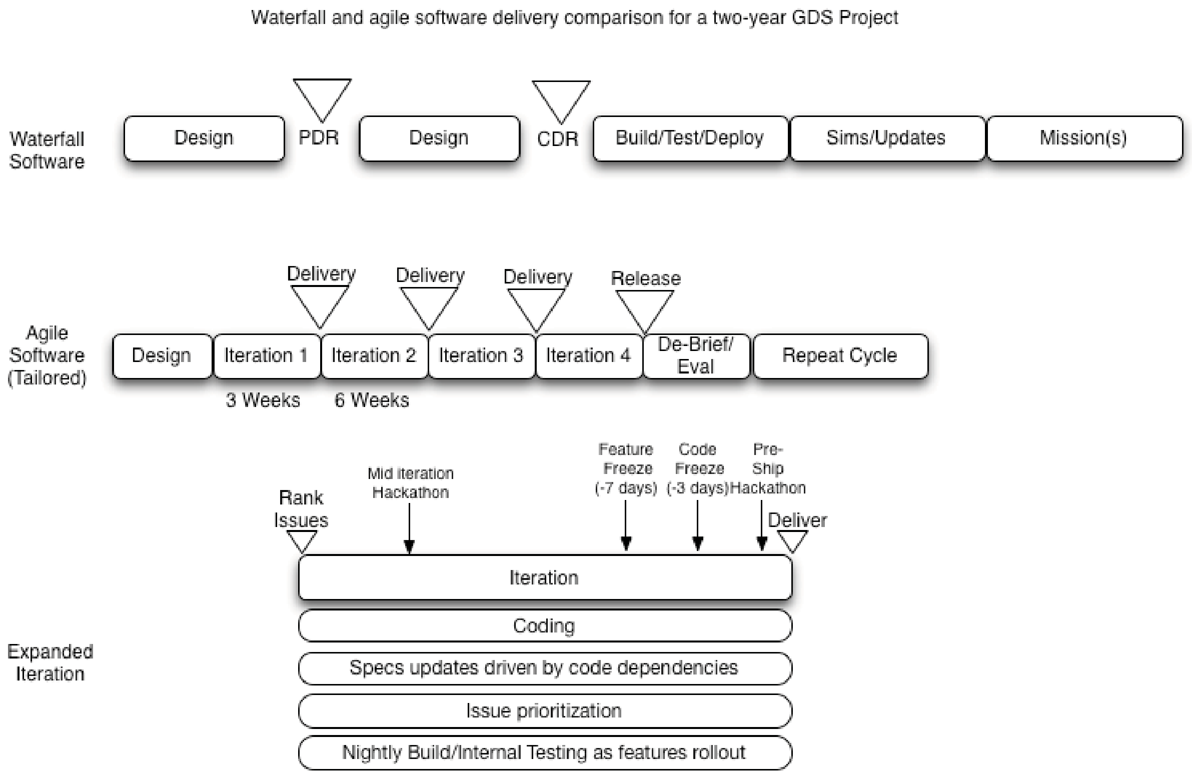

Figure 1. Waterfall vs. tailored Agile software development

B. Lean Methods

Lean methods for UX Design[4] minimize waste with a set of methods for testing proposed solutions for suitability using minimal resources to test those solutions. Based on the results, team resources are focused on the right solution to the most important problems.

We utilized these LeanUX principles:

- Removal of waste – anything that does not lead to the ultimate goal is waste
- Small batch size
- Continuous discovery
- Shared understanding
- Externalizing your work
- Making over analysis
- Outcomes, not output
- Outcomes over artifacts

The principles above are a subset of the principles of LeanUX [4]. Let's look briefly at some of these principles, then apply them to the specific context of RPM mission operations. Removal of waste comes from lean manufacturing. The goal is simple: to focus resources where they are most effective in creating the desired outcome(s). Externalizing your work means to make it visible (Figure 3). This aids in creating shared understanding. Making over analysis means simply that there is higher value in making something than in debating it. The team's time is better spent making part of the product than debating in meetings. The remaining two principles – outcomes not output, and outcomes over artifacts, speak to the same goal – focus your limited resources to get the outcomes you want. For an MOS, the outcome is the capability to perform the mission safely and effectively. We'll look at the specific context of RPM, then discuss the application of agile and lean in the design and development of the MOS.

Lean methods are based around using testable hypotheses to determine the suitability of proposed solutions. In market-driven software development, the application of lean methods would involve testing a hypothesis by quickly developing a minimum viable product (MVP) to test, often in the form of a simple web page to mock up a product. This is typically done in days. The practical implication is that resources are focused on solving the right problems.

Now let's look at these principles in the context of the RPM Mission.

III. RPM Operational Context

To understand the use of agile and lean methods for RPM mission operations we need to understand the operational requirements, the operational concepts and assumptions, and the known state of both the problems and the solutions, then define the areas that require design, testing, and validation.

RPM is a lunar prospecting mission. The primary mission goal is the detection and characterization of volatiles in a lunar polar region. The Lunar Crater Observation and Sensing Satellite (LCROSS) verified the existence of volatiles at the lunar poles using remote sensing. RPM follows that mission with in-situ measurements. The payload includes prospecting instruments, a drill to collect samples, and in-situ resource utilization (ISRU) experiments.

The combination of short mission duration, near real time command and control, and ambitious mission goals make for an operationally challenging mission. The duration is currently planned to range from five to seven days. During that time we plan to prospect, collect samples, and process those samples using the ISRU payload. The traverse plan has the rover going into permanently shadowed regions (PSRs). A PSR may not have seen light for hundreds of millions of years and the characterization of the surface for driving is largely unknown. Lunar operations are near real time. The cruise time to the moon is measured in days, not months, meaning the team has to be trained and ready for operations by launch. RPM is a fixed-budget mission. The expectation is that the MOS and GDS will be designed, built, and operated at low cost. The combination of challenging short-duration operations and a low budget presents an opportunity to use innovative methods for the system design, hence the application of agile and lean to the whole mission operations system.

IV. Designing the MOS

Table 1. Mission Operations System Design Inheritance

MOS Element	Design	Inheritance	Design method	Comments
Team Composition, Roles, Responsibilities	Team composition, roles, responsibilities	Precedents from human and robotic spaceflight	Design, sim, de-brief, iterate (DSDI)	Design pattern is inherited, specific instantiation is new
Decisions	Design for mission	Established near-real time mission decision structure	DSDI	Mission-specific decisions are new design
Voice Protocols	Standard protocols, no new design	Standard protocols	Inherit protocols	
Voice Calls	Mission specific calls/decisions	Established real-time voice call patterns	DSDI	
Command Classifications	Mission-specific classifications	Real-time command	DSDI	Inherit structure of classifications, map mission-specific command to that structure
Planning Processes	Reactive near real time; tactical and strategic planning	Real-time + dual-shift replanning	Initial design, test, iterate and refine	Real-time reactive prospecting process is new design
Flight Rules	Flight rules	Real-time flight rules	DSDI	Inherit and tailor template; content is mission specific
Procedures	Mission procedures	Real-time procedures	DSDI	Inherit and tailor template; content is mission specific
Distributed Operations Plan	Fully distributed operations	Distributed real-time operations	Hypothesis, test, refine	RPM fully distributed operations is new

We refer to the MOS to mean the mission operations team, processes, procedures, protocols, operational plans, decisions, and flight rules. Each of these elements has a variable degree of inheritance based on precedents from previous missions, as summarized in Table 1.

The core elements of a mission operations system are well known. The structure of procedures, flight rules, voice protocols, decisions, and other elements for each mission are tailored from previous missions. For example, we have to design flight rules for RPM, meaning we have to develop the rules for this mission. We are not designing how to build flight rules, but rather the rules themselves.

The degree of new design required varies by element. For planning, we will have a three-tiered process of near real time reactive, tactical, and strategic planning. The near real time reactive robotic prospecting nature of this cycle is new. The interaction of the reactive process with the tactical and strategic planning for a 24-hour, seven-days-per-week asset on the Moon is new. RPM's baseline operational plan has all mission controllers operating from their home institutions. This will take distributed decision making to a new level. This approach is new and must be tested and validated to determine that it is feasible.

V. Application of Principles to Mission Operations System (MOS) Design

Now let's look at the key tenets of the RPM MOS development plan and see how they relate to agile and lean principles.

Say it, then Sim it

As with most missions, some parts of the mission operations system design spark controversy within the team. While spirited debate fueled by passion for mission success is encouraged, we seek a means to resolve issues that goes beyond debates in meeting rooms. Here we use an analog to the lean principle of making over analysis, where the MVP substitutes for extended debate in conference rooms. Our method for testing solutions quickly will be to use paper simulations as a design and validation tool for operational designs. Rather than extended discussions in a series of meetings, we will first say it, then sim it.

Applicable principles from tailored agile and lean:

- Continuous discovery
- Shared understanding
- Making over analysis
- Outcomes, not output

Test Early and Often

We will constantly test our mission operations concepts using mockups, walkthroughs, and simulation. Early tests will be low-fidelity "paper" sims, with increasing fidelity as tools become available.

Applicable principles from agile and lean:

- Small batch size
- Continuous discovery
- Externalize your work
- Outcomes, not output
- Constant engagement
- Working software over comprehensive documentation

Participatory Design

Participatory Design (PD) is based on established principles that have been well documented [5,6,7]. We have previously applied participatory design to the design of both software and mission operations systems. This is a guided process in which the participants establish a common language of goals, triggers, outcomes, workflows, software objects, and interactions. PD will be used to design new elements and will be the basis of designs that we simulate.

Applicable principles from tailored agile and lean:

- Making over analysis
- Externalize your work
- Shared understanding
- Continuous discovery

Figure 2. Using participatory design to design RPM science operations by mission phase

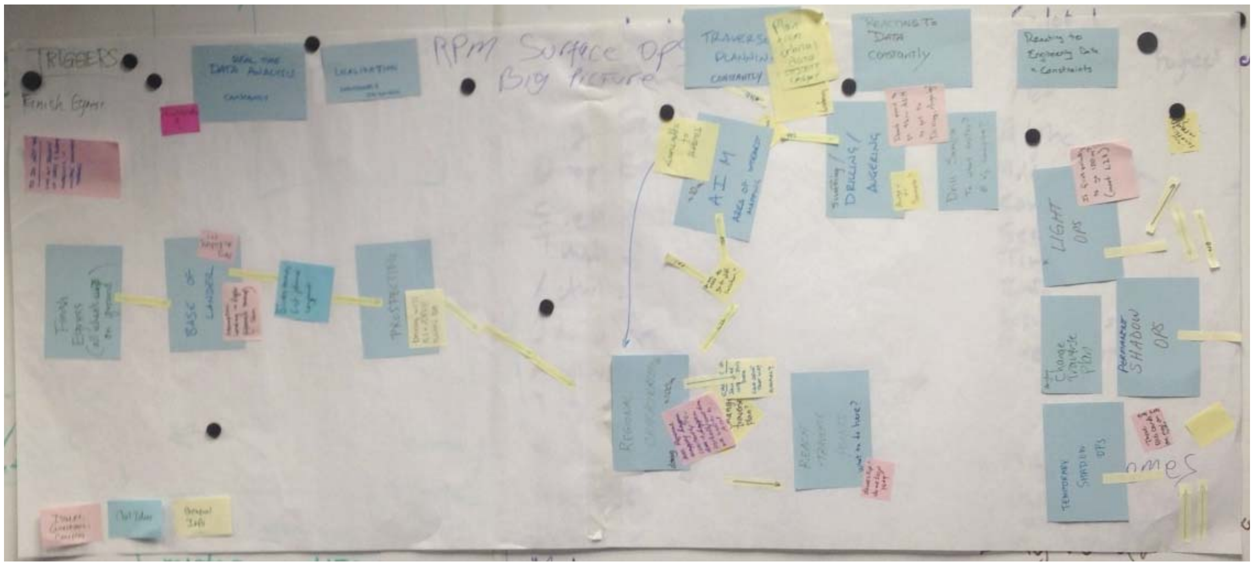

Figure 3. Externalizing our work – RPM high-level science operations by mission phase

Demonstrations, not Presentations

The goal here is to achieve project and review board acceptance on measures of progress, and to do so without extensive time devoted to putting together presentations. With each simulation, we will track metrics – time to completion, our ability to stay on our pre-planned timeline, did we complete the mission objectives, did we make errors – and these metrics will form the basis of our own understanding of our capability. This will be externalized as per lean principles and made available to the team and review boards. It is hoped that this can replace the need for extensive time spent developing presentations to show progress, especially late in the training flow when time is short and resources must be focused on the most important issues.

Applicable principles from tailored agile and lean:

- Working software over comprehensive documentation
- The measure of progress is working code
- Demonstrations, not presentations
- Rate of progress demonstrated through constantly exposing the work
- Outcomes over artifacts

Conclusion

Since the inception of RPM MOS just over a year ago, we have made substantial progress towards implementing the principles and methods described here. The initial PD sessions have produced results, built shared understanding, and have been accomplished in less time than traditional methods. We are within weeks of conducing our first paper simulation. Meeting time has been minimized. Issues that could become controversial are put on the list to validate in simulations. Our fully distributed operations plan awaits validation in simulations. The application of agile and lean methods to the full MOS system design for RPM is in the early stages. Based on results to date and our related experience in software design, we expect to be able to focus our limited resources effectively on what matters most – the outcome of maximizing the use of our mission assets on the Moon to achieve mission goals.

References

1. Poppendieck, M. and T. http://www.poppendieck.com/people.htm
2. Agile Manifesto, http://agilemanifesto.org/
3. Trimble, J. and Webster, C. "Agile Development Methods for Space Operations." Spaceops 2012
4. Gothelf, J. and Seiden, J. *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly, 2013.
5. Participatory GUI Design from Task Models, Tom Dayton, Joseph Kramer, Al McFarland, Monica Heidelberg, CHI 96
6. Bridging User Needs to OO GUI Prototype Via Task Object Design, Tom Dayton, Al McFarland, Joseph Kramer, CRC Press, 1998
7. The Democratization of Mission Control, Empowering Users, Jay Trimble, Tom Dayton, Alan Crocker, CHI 2013

Acknowledgements

Sarah Hobart, Tom Dayton, Madelyn Quinol, Sue Blumenberg