SAN FRANCISCO BAY UNIVERSITY

NASA'S VIPER LUNAR ROVER AGILE MISSION SYSTEM

Individual Project

Agile Project Management
MGT 501
Summer 2024

Persabel Y. Tesfay
20127

Monday, July 15th, 2024

Purpose of Analysis

The primary purpose of this analysis is to evaluate the incorporation of Agile methodologies into the development and operational phases of NASA's VIPER mission. This evaluation focuses on understanding how Agile practices have been applied to enhance the mission's effectiveness, even before the rover's launch. By examining the progress made, the benefits observed, and the potential for future success, this analysis aims to demonstrate the value of Agile methodologies in managing complex space missions. It also seeks to illustrate that ongoing projects can provide significant insights into the efficacy of Agile practices, making a case for their broader applicability in future NASA missions and other high-stakes endeavors.

Introduction

The Volatiles Investigation Polar Exploration Rover (VIPER), set to launch in November 2024, is a solar-powered rover developed by NASA's Science Mission Directorate (SMD) Planetary Science Division to search for volatiles at the lunar south pole. VIPER's mission is operationally complex, operating in challenging lighting and communication environments, and requires new designs in areas such as mission planning and real-time waypoint driving. The Mission System will manage VIPER during its cruise and surface operations on the moon. To enhance the efficiency and flexibility of the Mission System design, and subsequently improve its performance and reliability, Agile principles have been effectively applied to the ground data system software development and the design of mission system elements.

Trimble (2023) defines mission system (MS) to be the combined mission operations system (MOS) and ground data system (GDS). The MOS is the team, operational products and processes needed to operate the mission. The GDS is the hardware, software and facilities used by the MOS to operate the spacecraft (p. 1).

Project Mission

VIPER's mission goals are to characterize the distribution and physical state of lunar polar water and other volatiles in lunar cold traps, and regolith, and provide the data necessary for NASA to evaluate potential return of In-Situ Resource Utilization (ISRU) from lunar polar regions (Ennico-Smith et al., 2022, p. 1). To operate, VIPER requires line of site communication to Earth, as well as Sun on the solar panels. The lighting environment at the lunar south pole is highly dynamic, with shadows moving at roughly the rate of the rover. The mission operates on tight time margins to accomplish the science objectives while operating in both communications and sun line of site and staying ahead of the shadows. The operations design, both for mission planning and real-time operations, is complex and challenging.

Rationale for NASA's Adoption of Agile Methodology

NASA' s past experience with software development and mission system engineering for missions transitioned from a conventional waterfall approach to incorporating aspects of Agile methodology, and eventually embraced a user-centered Agile methodology. This transition was driven by several key factors, including:

- **Length of the Cycle:** The traditional six-month cycle for requirements definition, development, testing, and delivery was too long, leading to a large number of requirements that needed to be implemented over each cycle.
- **Increased Complexity:** The long cycles complicated design, development, and testing due to the large number of requirements.
- **Deferred Tasks:** Difficult and complex development tasks were sometimes deferred over the long cycle.

- **Ineffective Progress Metrics:** Traditional measures like lines of code were used as progress metrics. These metrics were easy to measure internally but did not translate to meaningful demonstrations of progress either internally or externally.
- **Expensive Bug Fixes:** Testing six months of new code at the end of a cycle meant that any bugs found were expensive to fix due to the extended time gap.
- **Pressure to Ship:** There was always strong pressure to ship since the customer had been waiting for months, leading to minor issues being resolved with workarounds.
- **Delayed Issue Resolution:** Customers had to wait another six months for the resolution of issues found in acceptance testing.
- **Divergence from Customer Expectations:** The long six-month period from specification to delivery caused the development team to fall out of touch with customers, creating a mismatch in expectations.
- **Customer Disappointment and Frustration:** The mismatch in expectations resulted in disappointment and frustration for both the developers and the NASA colleagues receiving the software.

Prior to adopting Agile methods, projects were evaluated using metrics and analysis, but the working code could not be tested until the final stages of testing and delivery. With modern software techniques, nightly or continuous builds were implemented, allowing the code's functionality to be tested on a daily basis. Thus, this progression stemmed from the need to solve problems encountered during development, rather than from a deliberate effort to conform to a specific development paradigm.

Plans, Processes and Strategies Used

A. Software development

Envision Phase

During the envision phase of software development for the VIPER mission, the focus was on defining clear objectives and requirements that aligned with NASA's mission goals. This phase involved:

- **Project Goals and Scope:** The primary goal of the software development was to create applications for navigation, communication, data analysis, and user interface that would aid in mapping different thermal environments on the lunar surface. The scope included developing software that could handle real-time data processing, provide accurate navigation and positioning, facilitate seamless communication between the rover and mission control, and deliver user-friendly interfaces for mission operators.
- **Requirements Gathering:** Engaging stakeholders to gather detailed requirements, including user interface specifications (like wireframes for human interfaces) and system-level requirements (down to level four).
- **Risk Assessment:** Identifying potential risks associated with software development, such as integration challenges between different software modules and ensuring compatibility with the mission's hardware.

Speculate Phase

The team speculated on various approaches to reduce the delivery cycle, starting with industry standards and best practices. They explored the concept of shortening the delivery cycle from a traditional lengthy process to a more frequent and predictable schedule. Initially, the cycle was reduced to six weeks and then further shortened to three weeks, aligning with typical Agile sprints.

- **Detailed Requirements Analysis:** High-level requirements were broken down into detailed specifications for each software application, focusing on their role in thermal environment mapping. Additionally, use cases and user stories were developed to illustrate specific functionalities and interactions, ensuring that all stakeholder needs were accurately captured.

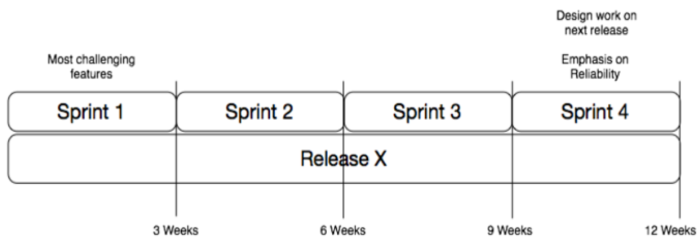

Figure 1. Four Sprint Release Cycle

Explore Phase

During the explore phase, the MOS team established a continuous integration and delivery approach to facilitate rapid development and frequent feedback. Each sprint included a develop/test/deliver cycle, with four sprints constituting a release (figure 2). This iterative approach was chosen to ensure that bugs could be found and fixed more efficiently during unit testing. A three-tiered structure for interaction with the customer:

- **Nightly or Continuous Builds:** A continuous integration and continuous delivery (CI/CD) pipeline was established to automate the building and testing of software. These nightly builds were made available for testing and feedback, allowing stakeholders to download and evaluate the latest software versions daily.
- **Three-Week Sprints:** Development work was organized into three-week sprints, with each sprint focusing on delivering specific features or improvements that the customer tested and either accepted or rejected.

- **Releases:** After four sprints, a release was intended for deployment into the mission operations environment.

These practices ensured that progress was measured by working code rather than presentations, facilitating immediate feedback and validation.

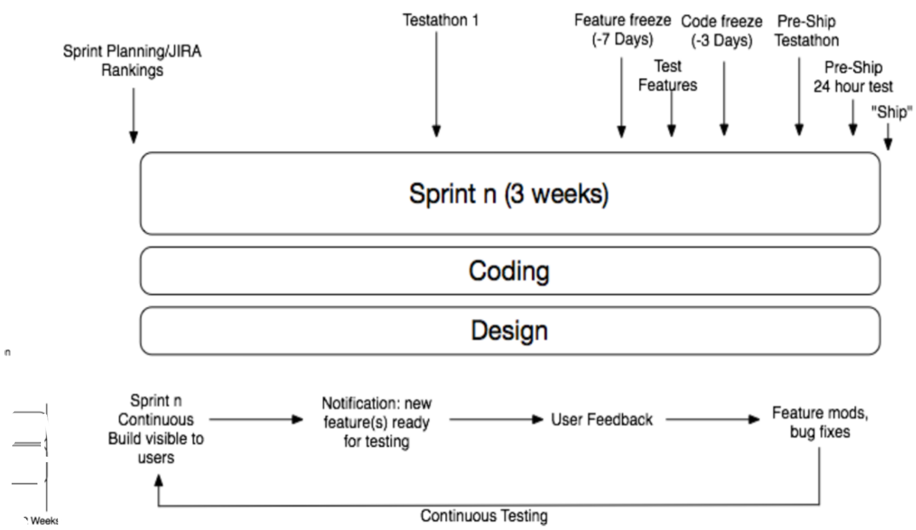

Figure 2. Agile Sprint Detail

Adopt Phase

The adopt phase involved implementing feedback mechanisms and enhancing collaboration where features were adjusted and refined based on feedback received during nightly or continuous builds and three-week sprints, ensuring that the software met user needs and expectations.

Close Phase

The move to Agile addressed the most pressing development and delivery problems. The short delivery cycles created a manageable set of features to develop and test within each sprint. The team was able to show demonstrable progress frequently, improving internal and customer satisfaction. The predictable and rapid delivery schedule ensured that features were delivered on time, and any issues could be addressed promptly in subsequent sprints. The enhanced interaction and feedback loops resulted in a successful project closure, with improved software quality and customer satisfaction.

B. Mission systems

On VIPER MS, in addition to using agile for the GDS, they also applied agile to the heart of MOS development, including, but not limited the following activities:

- Development simulations
- Operational Product Development
- Test and Training

Envision Phase

The VIPER Mission System's transition to Agile methodologies was driven by the need for a more efficient and flexible approach to handle the complex and dynamic nature of lunar missions. The fundamental requirements included:

- Developing simulations early in the mission lifecycle to influence both space and ground segment designs.
- Ensuring that operational product development and training/testing phases were iterative and closely linked to the mission's evolving needs.
- Creating a system that could handle real-time operations, taking into account the challenging lighting and communications environment at the lunar south pole.

Speculate Phase

Mission operations systems processes did not utilize nightly builds like the software development required. Instead, capabilities were showcased directly through different approaches, including procedure walkthroughs, lab-based development simulations, and mission simulations in a control center using the mission software with a simulator. During this phase agile principles were tailored to fit for mission systems as following:

- Assessment of capability through demonstrated execution of mission capability
- Maturation, evaluation, and iterative development of the system through continuous use during development
- Early and frequent builds and tests

Explore Phase

In the explore phase for mission systems, various concepts and technologies were investigated to determine their feasibility and effectiveness. This phase involved rigorous testing and evaluation of different approaches to identify the most suitable solutions for the mission.

- **Development Simulations (Dev Sims):** VIPER's dev sims focused on real-time driving simulations, identifying potential issues such as aft camera system and sunward driving problems. These simulations helped refine the rover's design and operational strategies, especially in terms of speed made good (SMG) for mission planning.
- **Say It Then Sim It:** This technique involved using simulations to resolve team differences and add data to discussions, ensuring that design decisions were data-driven.
- **Operational Product Development:** Agile ops product development involved iterative sprints aligned with the engineering and test/training schedules. Smaller, frequent drops replaced the traditional large drops, ensuring continuous improvement and adaptation.
- **Test & Training:** Structured around progressively testing and training by executing mission activities based on the mission timeline and surface traverse plan. Each test and

training session included debriefs and updates, feeding back into the software and ops products.

Adopt Phase

The adopt phase for mission systems development in the VIPER mission focused on implementing architectural designs and integration plans formulated during the envision phase. This phase involved:

- **Integration and Testing:** System integration efforts intensified, with a focus on assembling subsystems and components according to the predefined architecture. Integration testing sessions were conducted to verify the functionality and interoperability of mission systems under simulated lunar conditions. This iterative process allowed engineers to identify and resolve integration issues proactively.
- **Deployment Preparation:** Plans for deploying mission systems in the lunar environment were finalized during the adopt phase. Contingency plans were developed to address potential deployment challenges, such as communication latency, power management, and environmental hazards. Mission readiness reviews were conducted to assess the overall preparedness of mission systems for deployment and operational use.

Close Phase

During the close phase for mission systems in the VIPER mission, the focus shifts towards validating and verifying to ensure that mission systems met performance requirements and safety standards. Validation tests simulated operational scenarios to assess system reliability, resilience to environmental factors, and adherence to mission objectives. Verification activities confirmed that system outputs and behaviors aligned with predefined specifications and operational goals.

The iterative nature of Agile ensured that by the end of the project, the VIPER Mission System was well-tested and refined. Continuous feedback loops and the sprint structure enabled the team to make rapid adjustments, ensuring the system was ready for the operational challenges of the lunar mission.

Challenges

Resistance to Agile Adoption

NASA, with its long history of using traditional waterfall methodologies, encountered significant resistance to the adoption of Agile methodologies. The established processes and workflows were deeply ingrained, and shifting to a more flexible and iterative approach required a cultural change within the organization. Convincing stakeholders of the benefits of Agile and ensuring their buy-in was a considerable challenge. Additionally, the mindset shift from a sequential, phase-driven approach to Agile's iterative and collaborative model was met with skepticism among veteran team members.

Lack of Management Support

Initially, there was a lack of strong management support for Agile practices. Senior management, accustomed to traditional project management methods, were skeptical about the benefits of Agile and concerned about the perceived lack of rigor and documentation. Securing their commitment to Agile practices was essential for the successful implementation of these methodologies. Overcoming this challenge involved demonstrating how Agile could enhance adaptability, responsiveness, and efficiency without compromising on quality or safety, crucial for NASA's mission-critical projects.

Communication and Collaboration Challenges

Effective communication and collaboration are critical to the success of Agile projects. NASA faced challenges in fostering an environment where team members could openly communicate, share ideas, and work collaboratively across different disciplines and project phases. Ensuring consistent commitment to Agile practices across all teams and stakeholders was also challenging, given the diverse and dispersed nature of NASA's project teams. Aligning goals, expectations, and methodologies among stakeholders required continuous effort and proactive communication strategies.

Adjustments to Team dynamics

Building effective Agile teams required adjustments to team dynamics. The close-knit, cross-functional nature of Agile teams was a departure from the more siloed approach of traditional methodologies. Ensuring that team members could work together effectively, resolve conflicts, and maintain high levels of productivity and morale was a significant challenge.

Balancing Documentation with Agile flexibility

NASA's traditional, plan-driven methods, such as the Capability Maturity Model Integration (CMMI) required for critical software projects, emphasize rigorous documentation and process adherence. Balancing these requirements with the more flexible and iterative Agile methods was challenging. The need for extensive documentation and proof of process compliance sometimes conflicted with the Agile principle of minimal documentation.

Challenges in Software Development and Integration

The VIPER mission faced additional challenges in software development and integration. The VIPER MS wrote detailed requirements documents down to level four, including specifications like wireframes for software with a human interface. Verification processes tested the system against these requirements, while validation processes determined if the system effectively solved intended problems, simulated by the mission operations team. Attempting complex integrations over months presented significant issues, as separate development tracks for the rover and mission systems using Agile methods often led to compatibility and functionality issues during major system deliveries.

Lessons Learned

Importance of Cultural Change and Buy-in

NASA learned that shifting from traditional waterfall methodologies to Agile requires a significant cultural change. Convincing stakeholders, including senior management and veteran team members, of the benefits of Agile methodologies is crucial. It necessitates proactive communication, demonstrating how Agile can enhance adaptability, responsiveness, and efficiency without compromising on quality or safety.

Enhanced Communication and Collaboration

Effective communication and collaboration are fundamental to Agile success. NASA realized the importance of fostering an environment where team members across different disciplines and project phases could openly communicate, share ideas, and work collaboratively. Continuous effort in aligning goals, expectations, and methodologies among stakeholders proved essential in mitigating misunderstandings and improving project outcomes.

Adaptation of Team Dynamics

Building effective Agile teams requires adjustments to team dynamics. The shift from siloed, specialized roles in traditional methodologies to cross-functional Agile teams was challenging but necessary. NASA focused on promoting teamwork, resolving conflicts promptly, and maintaining high levels of productivity and morale among team members.

Balancing Rigor and Flexibility

NASA encountered the challenge of balancing rigorous documentation and process adherence, required by traditional methods like Capability Maturity Model Integration (CMMI), with Agile's emphasis on minimal documentation and iterative development. Finding the right balance was essential to meeting regulatory and safety standards while leveraging Agile's benefits in enhancing project flexibility and responsiveness.

Integration and Testing Challenges

The VIPER mission highlighted the complexities of integrating separate Agile development tracks for the rover and mission systems. Attempting major system integrations over extended periods led to compatibility and functionality issues. NASA learned the importance of more frequent integration points and continuous testing throughout the development lifecycle to identify and address issues early.

Continuous Improvement and Adaptation

Above all, NASA emphasized the need for continuous improvement and adaptation in Agile practices. Learning from challenges, adjusting methodologies, and implementing lessons learned from each project iteration were critical. This iterative approach ensured ongoing enhancement of processes, team capabilities, and overall project outcomes.

Organizational Integration and Continuous Integration

NASA identified an organizational divide where separate Agile development tracks for the rover and mission systems led to integration challenges. To address this, NASA implemented weekly software integrations across rover and mission systems. This approach allowed for continuous problem-solving and testing using the software, preventing issues from accumulating over time. By adopting this practice, NASA improved collaboration between teams and reduced integration-related risks during major system deliveries.

Reflecting on Agile Implementation in the VIPER Lunar Rover Project

The VIPER project by NASA represents a significant advancement in lunar exploration, aiming to explore the lunar south pole for volatiles. The integration of Agile methodologies into such a complex mission highlights NASA's adaptability and commitment to improving project efficiency and flexibility. The project faced notable challenges, particularly in organizational integration and software development, which required continuous adjustments and proactive management.

Agile Project Management Principles Used:

- **Iterative Development:** NASA used iterative cycles to develop and integrate rover and mission systems software, allowing for continuous improvement and adaptation.
- **Collaboration and Communication:** Agile principles encouraged cross-functional teams to collaborate closely, enhancing communication and problem-solving capabilities.
- **Adaptability:** Agile's flexibility enabled NASA to respond to changing requirements and challenges in real-time, crucial for a mission with dynamic operational environments.

These principles were tailored for this mission's software development and mission systems by emphasizing key aspects of the Agile Manifesto, such as prioritizing individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. This tailored approach included:

- **Assessment of Capability Through Demonstrated Execution of Mission Capability:** Utilizing real-world scenarios and simulations to validate the software's functionality and effectiveness in achieving mission goals.

- **Continuous System Maturation and Iterative Development:** Emphasizing effective communication and collaboration among team members, engaging stakeholders continuously, and maintaining flexibility to adapt to new information and changing requirements throughout the project's lifecycle.
- **Early and Frequent Builds and Tests:** Prioritizing the delivery of functional software that can be tested and iterated upon over extensive documentation. Implementing early and regular software builds and tests to identify and address issues promptly, ensuring the system's robustness and reliability.

Traditional or Waterfall Approach Suitability

A traditional or waterfall project management approach might not have been as suitable for the VIPER project due to its dynamic and exploratory nature. Waterfall's sequential phases and fixed requirements would likely have hindered NASA's ability to adapt to emerging challenges and evolving scientific objectives. Agile's iterative nature allowed for continuous refinement and adjustment, crucial for a mission where scientific discovery and technological challenges go hand-in-hand.

Recommendations and Suggestions

Enhanced Stakeholder Engagement: Increasing stakeholder involvement throughout all project phases can ensure that mission objectives align closely with scientific and operational needs.

Streamlined Documentation Processes: Further refining documentation practices to balance regulatory requirements with Agile's principles can enhance efficiency without compromising safety or quality.

Continuous Integration and Testing: Maintaining and expanding weekly software integrations across rover and mission systems can further reduce integration risks and improve overall system reliability.

Training and Skill Development: Continued training in Agile methodologies for all project teams, including SA personnel, can improve understanding and application, enhancing Agile's effectiveness within NASA's project frameworks.

In conclusion, while the VIPER project demonstrated the benefits of Agile methodologies in enhancing adaptability and collaboration, ongoing refinement and alignment with project-specific needs are crucial for future missions. NASA's commitment to learning from these experiences will undoubtedly contribute to more efficient and successful missions in the future.

Conclusion

In conclusion, the ongoing preparation of NASA's VIPER mission demonstrates the promising application of Agile methodologies in advancing lunar exploration capabilities. By integrating Agile principles such as iterative development, collaboration, adaptability, and early testing, NASA has laid a strong foundation for optimizing project efficiency and responsiveness. Despite initial challenges in cultural adoption and organizational alignment, the shift from traditional waterfall methods to Agile has enabled NASA to better manage the complexities of software development and mission system engineering for lunar exploration.

The VIPER project serves as a testament to Agile's effectiveness in mitigating risks and enhancing stakeholder engagement throughout the mission lifecycle. Moving forward, NASA's continued refinement of Agile practices will be crucial in ensuring readiness for the rover's eventual launch and operational success on the lunar surface. By leveraging insights gained from

VIPER, NASA is poised to further enhance its capabilities in managing future missions with agility and precision, paving the way for continued advancements in space exploration.

References

1. Ennico-Smith, K., Colaprete A., Lim, D.S.S., & Andrews, D. (2022, June 07-10). *The VIPER Mission, a Resource-Mapping Mission on Another Celestial Body* [Conference presentation]. SRR XXII Meeting Colorado School of Mines, Golden, Colorado, United States. https://shorturl.at/gQAvv
2. National Aeronautics and Space Administration. (2014, September). *Agile Development Brings New Challenges for Software Assurance at NASA.* Office of Safety and Mission Assurance. https://shorturl.at/cF2h9
3. Trimble, J. & Webster, C. (2012, June 11-15). *Agile Development Methods for Space Operation* [Conference presentation]. 12th International Conference in Space Operations, Stockholm, Sweden. http://dx.doi.org/10.2514/6.2012-1264554
4. Trible, J. (2014, May 05-09). *Lean Mission Operations Systems Design - Using Agile and Lean Development Principles for Mission Operations Design and Development* [Conference presentation]. 13th International Conference on Space Operations, Pasadena, California, United States. http://dx.doi.org/10.2514/6.2014-1816
5. Trible, J., Shirley, M., & Hobart, S. (2016, May 16-20). *Agile: From Software to Mission System* [Conference presentation]. 14th International Conference on Space Operations, Daejeon, Korea. http://dx.doi.org/10.2514/6.2016-2477
6. Trible, J. (2023, March 06-10). *VIPER Lunar Rover Agile Mission Systems* [Conference presentation]. 17th International Conference on Space Operations, Dubai, United Arab Emirates. https://shorturl.at/IGf00

7. Webster, C., Shi, N., & Smith, I. (2012, March 03-10). *Delivering Software into NASA's Mission Control Center Using Agile Development Techniques* [Conference presentation]. 2012 IEEE Aerospace Conference, Big Sky, Montana, United States. https://doi.org/10.1109/AERO.2012.6187329