

Activity 1 - Key Considerations

Alex M. Frear

College of Science, Engineering, and Technology, Grand Canyon University

Course Number: CST-326 Written and Verbal Communication for Software Development

Professor Stephanie Macuga

03/21/2025

Table of Contents

<i>What Is the Purpose of Scrum?.....</i>	<i>3</i>
<i>What Are the Key Considerations Necessary to Successfully Complete a Project Using Agile Project Management?</i>	<i>4</i>
<i>How Are Agile-Based Tools and a Git Repository Used to Manage Scrum?</i>	<i>5</i>
<i>References</i>	<i>6</i>

What Is the Purpose of Scrum?

Scrum is an agile framework designed to simplify the complex process of developing and delivering products through iterative progress, collaboration, and continuous improvement. Its primary purpose is to enable teams to adapt quickly to change by delivering working increments of a product in short, defined periods known as sprints. According to the Scrum Guide, “Scrum is a lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems” (Schwaber & Sutherland, 2020, p. 3). This approach breaks projects into manageable pieces, ensuring that feedback is incorporated regularly and that priorities can be adjusted based on evolving stakeholder needs.

At its core, Scrum defines clear roles: the Product Owner sets the vision and prioritizes the backlog, the Scrum Master facilitates the process and removes impediments, and the Development Team collaborates to deliver the product. As noted by Atlassian (2021), “Scrum promotes a mindset of collaboration and continuous improvement, ensuring every team member is aligned with the project’s objectives” (p. 1). Daily stand-ups, sprint reviews, and retrospectives foster open communication and ongoing learning, which are vital for addressing challenges promptly and refining processes.

Scrum’s iterative nature helps teams manage uncertainty and complexity. It creates an environment that not only emphasizes efficiency and productivity but also nurtures creativity and innovation in problem-solving, thereby continuously adding value to both the product and the organization. This iterative approach not only reduces the risk of major project failures but also encourages continuous feedback loops that facilitate early detection of issues, allowing teams to pivot or re-align priorities as necessary. By engaging every team member in the process and promoting transparency, Scrum fosters accountability and shared ownership of project outcomes, ensuring that every contribution is valued.

What Are the Key Considerations Necessary to Successfully Complete a Project Using Agile Project Management?

Successful Agile project management requires a blend of strategic planning, effective communication, and continuous adaptation to navigate the complexities of modern software development. Establishing a shared vision and clear project goals is essential; a well-prioritized product backlog serves as the roadmap that guides the team through iterative sprints. The Agile Manifesto emphasizes valuing “individuals and interactions over processes and tools” (Beck et al., 2001), which underlines the critical importance of effective communication and active stakeholder engagement throughout the project lifecycle.

Regular interactions, such as daily stand-ups, sprint planning sessions, and retrospectives, are key to maintaining transparency and ensuring that any issues are promptly addressed. Highsmith (2019) asserts, “Agile methods transform how teams communicate by emphasizing adaptability and continuous feedback” (p. 45). These recurring touchpoints promote an environment where open dialogue and immediate adjustments become the norm, thereby facilitating swift decision-making and alignment on project priorities.

Risk management is another vital consideration in Agile project management. Continuous testing and integration practices allow teams to detect and resolve issues early in the development process, reducing the likelihood of significant setbacks later. This iterative process, when combined with a culture that fosters trust and cross-functional collaboration, empowers team members to share expertise and innovate effectively. Maintaining consistent stakeholder involvement through regular feedback loops ensures that the project remains aligned with user needs and business objectives, ultimately steering it toward successful outcomes. By embracing these key considerations, Agile teams can deliver high-quality products efficiently while adapting to changing requirements and mitigating risks throughout the project lifecycle.

How Are Agile-Based Tools and a Git Repository Used to Manage Scrum?

Agile-based tools and Git repositories work together to streamline the Scrum process by enhancing communication, task tracking, and version control. Tools like Jira, Trello, or Asana offer visual boards that break down the product backlog into manageable tasks and user stories. Atlassian (2021) explains, “Agile tools streamline the workflow by providing real-time updates and progress metrics, such as sprint velocity and burndown charts” (p. 2). This transparency enables teams to monitor progress, adjust priorities, and ensure that all members remain informed about project status.

In addition to facilitating task management, these agile tools foster collaboration by centralizing communication and documentation. They enable team members to attach files, comment on tasks, and link related work items, which creates an environment where all relevant information is easily accessible. This level of organization is crucial for maintaining clarity and alignment across the team during rapid sprint cycles.

Simultaneously, Git repositories are essential for effective version control. Developers use Git to manage code changes, work on multiple branches concurrently, and merge contributions safely, thereby minimizing conflicts. As Chacon and Straub (2014) observe, “Git’s distributed nature allows for comprehensive version control and collaborative development, which is essential in fast-paced agile environments” (p. 98). This system documents every modification, enabling teams to track the evolution of the codebase and revert changes if needed.

Integrating Git with agile tools further enhances project management by linking commits and pull requests directly to specific user stories or tasks. This seamless integration supports automated testing and continuous integration processes, which uphold code quality and accelerate iterative development. Together, these tools embody the Scrum principles of transparency, collaboration, and continuous improvement, ultimately leading to a more efficient and adaptive software development process.

References

Atlassian. (2021). What is Scrum? A practical guide to Agile project management. Retrieved from <https://www.atlassian.com/agile/scrum>

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Retrieved from <https://scrumguides.org>

Beck, K., et al. (2001). Manifesto for agile software development. Retrieved from <https://agilemanifesto.org>

Highsmith, J. (2019). Agile project management: Creating innovative products (2nd ed.). Addison-Wesley Professional.

Chacon, S., & Straub, B. (2014). Pro Git (2nd ed.). Apress. Retrieved from <https://git-scm.com/book/en/v2>