



Community of Practice

IT Project Management

October 2023

Agile for non-software projects

Purpose

The purpose of this document is to explain how agile methods and principles can be used for non-software projects.

Note: Although some agile methods are described in this document, it is not an agile instruction manual or user guide. Those interested in learning more about agile should contact their supervisors for training opportunities.

Background

What is agile?

Agile is an approach to software development that emerged in the 1990s and is an alternative to the traditional software development approach known as waterfall.

Although the principles of agile were originally developed for software, agile principles and methods can be applied to non-software projects.

How to adopt an agile mindset

Create smaller, incremental goals

With agile, a project is broken down into smaller, incremental goals. Goals are established in terms of what can reasonably be accomplished in a shorter period, called a “sprint” or “iteration.”

- For example, instead of a single feasibility report at the end of the project, segment the report into multiple deliverables. Sample deliverables—that are the building blocks of a feasibility report—include:
 - Project objectives
 - Needs Assessment
 - Solution alternatives analysis
 - Organizational impact
 - Program impact
- And divide each of those deliverables into even smaller segments. Focus on the outcomes (benefits) instead of the outputs (deliverables).
 - Project objectives
 - Problem to be solved
 - Response to statutory requirements

- Requirements
 - Current and future state workflows
 - Needs assessment or functional requirements
 - Security requirements
 - System performance requirements
- Solution alternatives
 - Alternative A assessment
 - Alternative B assessment
 - Cost estimates
- Each segment becomes a goal. And the project team determines how many of those goals can be accomplished in the sprint iteration (e.g., within 2 weeks).
- With agile, the smaller the goal better (usually).

Embrace change

With agile, change is welcomed.

- Project priorities are not set or dictated solely by chronology - The project team comes together regularly to establish priorities for each sprint.
- Frequent feedback - Find practical, actionable ways to check that the solution (e.g., analysis) meets the project's needs. The more uncertain about the solution, the sooner to get feedback.
- Change course if the feedback suggests you're missing the mark - Fast and frequent feedback is a way to keep the cost of change low.

Work on what's most important

- Identify the most important goal and complete that before moving onto the next most important one.
- Constrain the intake of work – Paradoxically, sometimes the more work in progress, the less gets done. Too much work contributes to delays, quality issues, missed commitments, and stress.
- Once the solution is working, move onto the next goal / outcome.

Review progress daily

A daily stand-up meeting is a chance for the project team to discuss a project's progress at a high level. The meeting lasts 15 minutes and each participant reports on their accomplishments since the last stand-up meeting. It helps ensure everyone is aligned and knows what's going on.

A repeatable agenda is a great way to keep the daily stand-up meeting running smoothly. Three sample questions to ask each person at the stand-up:

- What did you work on yesterday?
- What are you working on today?
- What issues are blocking you? (Or what help do you need)?

With a daily stand-up, the project team should be able to reduce the frequency and length of other project meetings (e.g., project status meetings).

Produce the minimum viable product (MVP)

A minimum viable product (MVP) is a version of a product with just enough features to satisfy early customers and provide feedback for future development. *Minimum* means limited features, and *Viable* means the features are enough for the product to function properly.

- In software development, the MVP version might be a basic working version of an IT system component (e.g., an online enrollment screen).
- For a feasibility study, MVP might mean producing a draft feasibility study section with enough information to get early feedback from the document's audience. For example, high level process flows for the requirements analysis.

Prioritize quality

Quality is just as critical in agile as with other project management approaches.

- Teams sometimes mistakenly stress the *minimum* part of MVP to the exclusion of the *viable* part. The product delivered must be of sufficient quality to provide an accurate assessment of whether customers will use the product.
- Teams sometimes deliver what they consider an MVP, and then do not make any further changes to that product, regardless of feedback they receive about it.

Assign agile team roles

In each agile team, there are several important team roles. How many are needed will vary according to the size of the team and the complexity of your project. Please note this section describes agile project roles (not positions).

- Product owner - Has the authority to make key decisions and in particular, is responsible for making sure the team is working on the right items.
- Developer - Builds the product. They are responsible for programming, testing, and debugging the product. For a feasibility study, they are responsible for writing and revising the report.
- Team lead / scrum master - Facilitates daily stand-up meetings and oversees sprint planning meetings. They help to keep the team focused and avoid scope-creep. They manage sprint reviews and gather feedback, and they also remove roadblocks.
- Independent tester - Is there to catch any mistakes the team has made before delivering the project to a client. For a feasibility study, the tester role could be an editor who checks a work product before it is published.

Summary

Although agile is primarily associated with software development, many agile concepts can apply to non-software projects.

- Instead of a master plan with milestones, focus on smaller segments (sprints).
- Instead of focusing on outputs (deliverables), focus on outcomes (benefits).
- Instead of waiting for a milestone to be delivered in a month or two (or six), review and check daily the progress of the work packages.
- Instead of multitasking, limit the amount of work in progress (WIP) at any time.
- Instead of a single product manager, distribute responsibilities among more members of the team.