Revised Definition of Agile

# Introduction

The purpose of this assignment is to consider (1) when definitions may be necessary and (2) how to choose an appropriate amount of detail for the audience and purpose. To complete this assignment, I will identify an audience and purpose and then I will write three definitions: a parenthetical definition, a sentence definition, and an expanded definition.

# Audience and Purpose

The following definitions are intended for first- or second-year computer science students who have not yet gotten any work experience in software development. To work effectively as part of a software development team, computer science students will need to understand an approach to project management known as “Agile,” which is very common in the software development industry.

# Parenthetical Definition

Today, the agile methodology (iterative approach to project management) is ubiquitous in software development, because it solves many of the problems caused by more traditional approaches to project management.

# Sentence Definition

The agile methodology is an approach to project management that enables software development teams to quickly respond to changing requirements and user feedback (Atlassian, n.d.). The iterative nature of the agile methodology facilitates continuous improvement of processes and promotes collaboration within and across interdisciplinary teams (Atlassian, n.d.).

# Expanded Definition

## What is Agile?

As its name implies, the agile methodology is an approach to project management that enables software development teams to adapt to changing requirements and quickly respond to user feedback (Atlassian, n.d.). Thanks to its iterative nature, the agile methodology has many advantages when compared with more traditional approaches to project management.

## Why is the Agile Methodology better than Waterfall?

Before the agile methodology was developed, most software development teams followed an approach to project management called “waterfall.” The agile methodology is a response to the many shortcomings of that earlier approach to project management. Unlike the agile methodology, the waterfall approach to project management is a linear sequence of phases that ends with a single launch, which means that end users cannot provide feedback about their experience with the product until after development has completed (Radigan, n.d.).

In addition, the waterfall approach to software development makes cross-functional collaboration difficult, if not impossible. With the waterfall approach, projects are always passed forward from one phase to the next; just as water never falls up, projects are never passed back to a previous phase. In other words, each phase is completed without input from members of the team that specialize in other areas. Moreover, the completion of each phase is final (Atlassian, n.d.).

Figure 1 The waterfall approach to project management.
Source: Computer History Museum. (n.d.). What is software? https://www.computerhistory.org/makesoftware/exhibit/software-lab/

In contrast, the agile methodology is iterative, such that everyone involved in the project can communicate and collaborate with everyone else. In addition, each iteration (or version) can be produced quickly and released to the public, which means that the next iteration may benefit from user feedback. Hence, the agile methodology enables teams and their products to continuously improve (Atlassian, n.d.).



Figure 2 The agile approach to project management.
Source: Computer History Museum. (n.d.). What is software? https://www.computerhistory.org/makesoftware/exhibit/software-lab/

## How is the Agile Methodology implemented?

The agile methodology may be implemented using a wide variety of frameworks, including Scrum, Kanban, Extreme Programming, and others. We will focus on Scrum. Scrum has four “ceremonies” (routine meetings):

1. Sprint Planning
2. Daily Standup
3. Sprint Demo
4. Retrospective

### Sprint Planning

A sprint is an interval of time during which each member of a software development team commits to completing an amount of work. Sprint Planning is a ceremony conducted at the beginning of a sprint, during which tasks are prioritized and assigned. Different companies and teams may have sprints of varying lengths, but two-week-long sprints are common (Atlassian, n.d.; Radigan, n.d.).

### Daily Standup

During the Daily Standup meeting, each member of the team briefly reports in succession the work that they completed yesterday, the work that they plan to complete today, and anything that is blocking them from making progress on their tasks. The Daily Standup meeting is so named, because attendees stand rather than sit, as a means of encouraging brevity and discouraging lengthy conversation (Atlassian, n.d.; Radigan, n.d.).

### Sprint Demo

During a Sprint Demo, individual developers have the opportunity to share new features that they have built with the rest of the team. Sprint Demos promote transparency and may inspire collaboration (Atlassian, n.d.; Radigan, n.d.).

### Retrospective

Agile teams conduct a Retrospective meeting at the end of each sprint. During a Retrospective meeting, each team member reflects on the sprint and comments on what worked well and what could have been done better. In this way, agile teams continuously improve teamwork, processes, and ultimately the products that they build (Atlassian, n.d.; Radigan, n.d.).

# References

Atlassian. (n.d.). *Agile project management: How agile methodologies can work for your software team.* <https://www.atlassian.com/agile/project-management>

Atlassian. (n.d.). *Scrum: Learn how to scrum with the best of ‘em.* <https://www.atlassian.com/agile/scrum>

Atlassian. (n.d.). *The agile coach: Atlassian’s no-nonsense guide to agile development*. <https://www.atlassian.com/agile>

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for agile software development*. <http://agilemanifesto.org/iso/en/manifesto.html>

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Principles behind the agile manifesto*. <http://agilemanifesto.org/principles.html>

Computer History Museum. (n.d.). *What is software?* <https://www.computerhistory.org/makesoftware/exhibit/software-lab/>

Nyce, C. M. (2017, December 8). *The winter getaway that turned the software world upside down: How a group of programming rebels started a global movement*. The Atlantic. <https://www.theatlantic.com/technology/archive/2017/12/agile-manifesto-a-history/547715/>

Radigan, D. (n.d.). *Agile project management*. Atlassian. <https://www.atlassian.com/agile/project-management/program>