UNIVERSITY of VIRGINIA | DARDEN Business Publishing

UVA-OM-1711
Rev. May 9, 2022

Agile Development

What Is Agile?

A group of software developers created the Agile Manifesto in 2001.[1] While it is only 68 words long, it has impacted areas of software and product development, and arguably all project management, in groundbreaking ways. The manifesto essentially says the following:

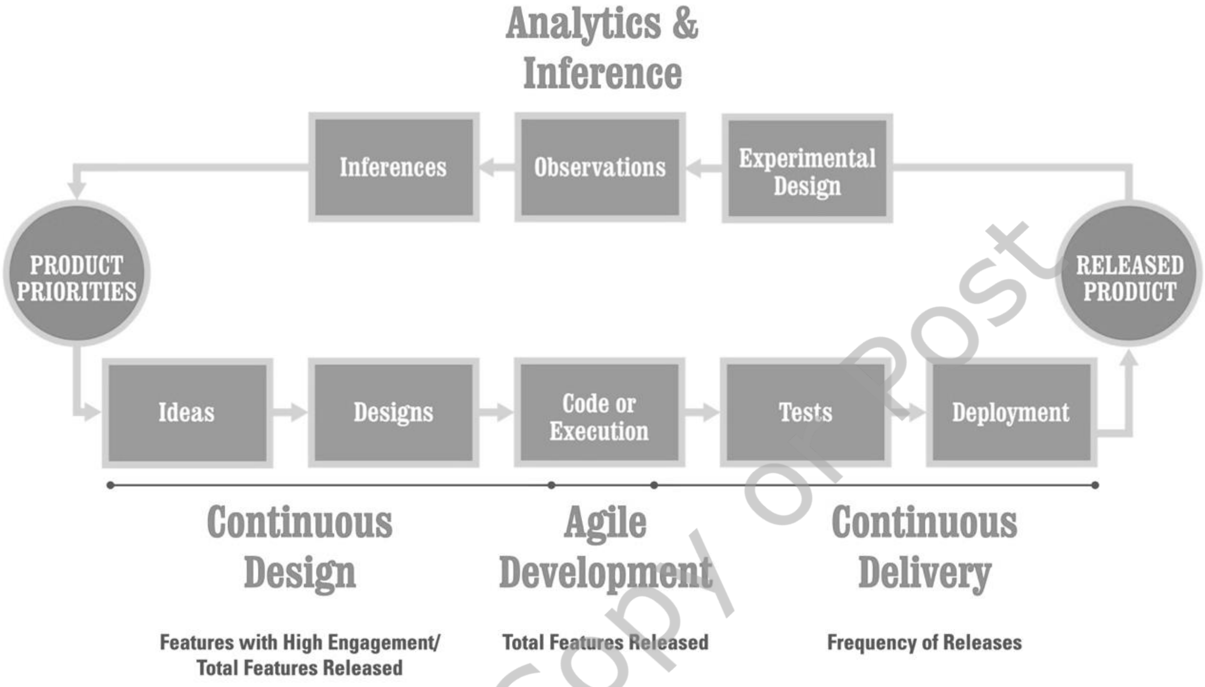
Individual interactions	over	processes and tools
Working software	over	comprehensive documentation (requirements)
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

That manifesto is probably the only single thing everyone would agree is agile. Agile is more a set of outcomes and circumstances that an individual team is seeking to realize than it is a specific practice or set of practices. For the general manager in digital, an actionable way to understand and approach agile is from the perspective of things their team does and things their team wants to achieve. **Figure 1** shows a product "pipeline," wherein a team goes from observation to working software.

[1] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, "Manifesto for Agile Software Development," 2001, https://agilemanifesto.org/ (accessed Dec. 1, 2020).

This technical note was prepared by Alex Cowan, General Faculty; and Yael Grushka-Cockayne, Professor of Business Administration.

Figure 1. Product pipeline.



Source: Unless otherwise specified, all figures were created by the authors.

Table 1 describes these areas of practice and how they relate to the various methodologies and practices associated with agile. It is worth noting that there are few universal terms in agile, and that while the terms below are common, they are not universal.

Table 1. Practice areas in agile development.

Area of Practice	Description	Prevalence	Related Areas and Terms
Continuous design, where success is improving on your ratios of features released to successful features	This area deals with figuring out *what* to build. While many agile teams consistently test usability, few run a consistent program for testing across problem diagnosis, demand evaluation, and interface usability. Success in this area means more of the content the team is releasing sees high engagement from customers or high performance from internal users.	Low/increasing	Product design Design thinking Customer discovery Customer development Lean Startup Lean user experience (UX) Lean analytics Design sprints **Legacy practices:** Requirements gathering Acceptance testing
Agile development, where success is improving on the	This area deals with building software applications and systems. Agile in this area is nearly a de facto standard.	High/steady	Scrum SAFE Kanban (loosely) XP

velocity of total features released	Success in this area means the team is improving its "velocity," as measured by the size and amount of features released in a given unit of time.		Pair programming Test-driven development Estimating Story points **Legacy practices:** Waterfall
Continuous delivery, where success is improving on the frequency of your releases	This area deals with integrating the jobs of testing and deploying software with the job of building it. Success in this area means the team is releasing more frequently. Continuous delivery means a developer can commit code and automatically (if it passes a series of tests) deploy it to users.	Moderate/ increasing	DevOps[2] Continuous integration Unit testing Integration testing System testing Staging Release management **Legacy practices:** Quality assurance
Hypothesis testing, where (like continous design) success is improving on your ratios of features released to successful features	This area, tightly related to continuous design and continuous delivery, deals with running experiments to drive to an evidence-based definition of "done." Success in this area means that teams assess whether they're "done" with an item based on directly observable user behaviors.	Low/increasing	A/B testing Experimentation

Source: Unless otherwise specified, all tables were created by the authors.

A general manager might reasonably ask: "What is agile, and how do I apply it?" Agile is not any single thing or practice, and yet it is also not subjective or abstract. An overly rigid view is that agile is whatever a particularly agile-related methodology like Scrum prescribes. An overly abstract view is that agile can only be understood as the principles described in the manifesto.

A balanced and purposeful view of agile for the general manager is that there is a product pipeline, it has certain segments (like the Areas of Practice in **Table 1**), those segments have certain observable success metrics, and agile is a body of practice that can help the manager improve on those success metrics. The challenge for the general manager is to select, test, and iterate on their particular practice of agile. This is the topic of the note's closing section, "How Do You Agile? Beyond the Basics." First, it's useful to understand the context of agile: when and why it's useful and the current state of practice in industry.

[2] DevOps is a "compound of development (Dev) and operations (Ops)." See "What Is DevOps?," Amazon Web Services, https://aws.amazon.com/devops/what-is-devops/ (accessed Dec. 10, 2020).