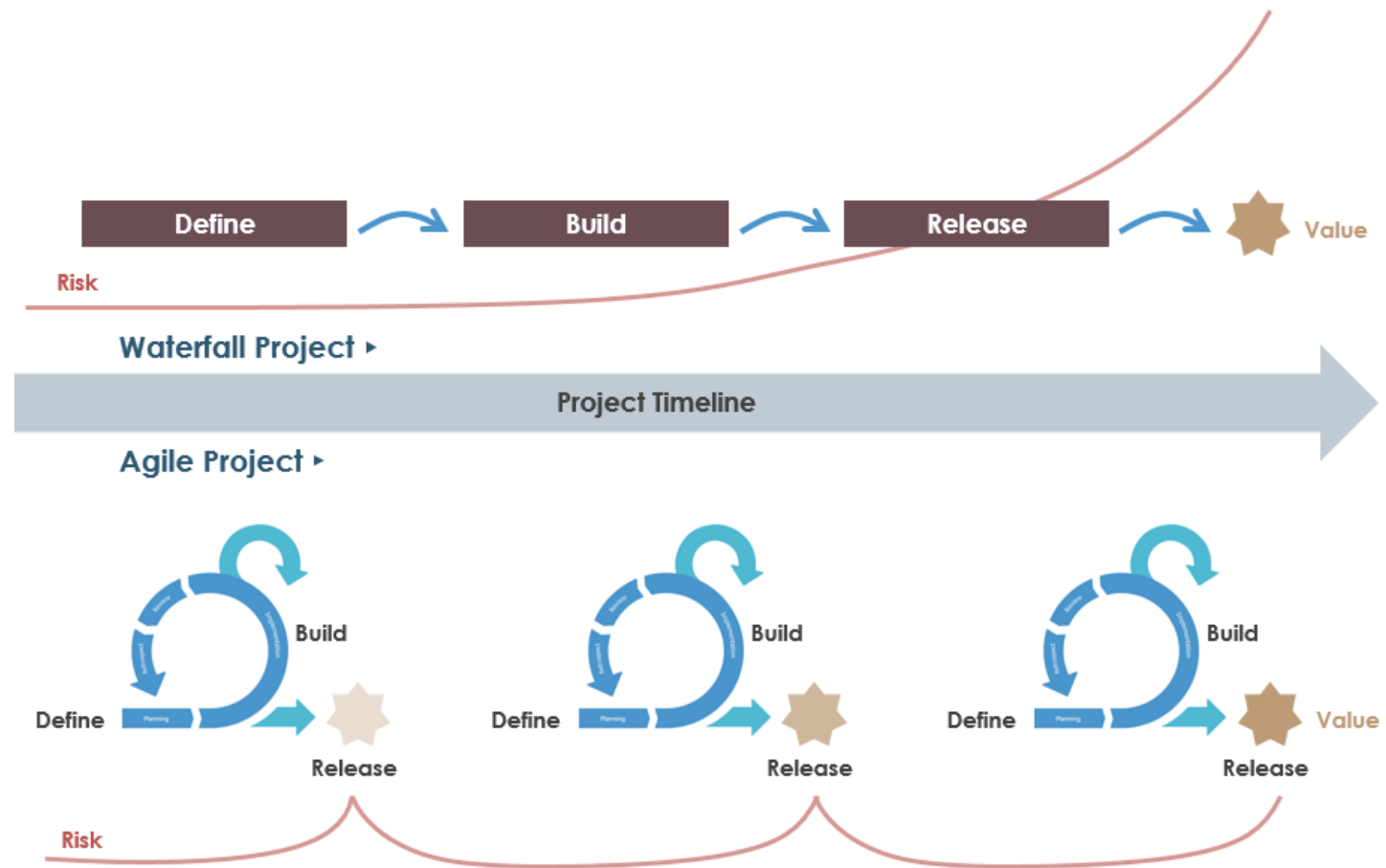


What is Agile Software Development?

Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning, instead of trying to deliver it all at once near the end.

Agile focuses on keeping the process lean and creating minimum viable products (MVPs) that go through a number of iterations before anything is final. Feedback is gathered and implemented continually and in all, it is a much more dynamic process where everyone is working together towards one goal.





Agile Software Development



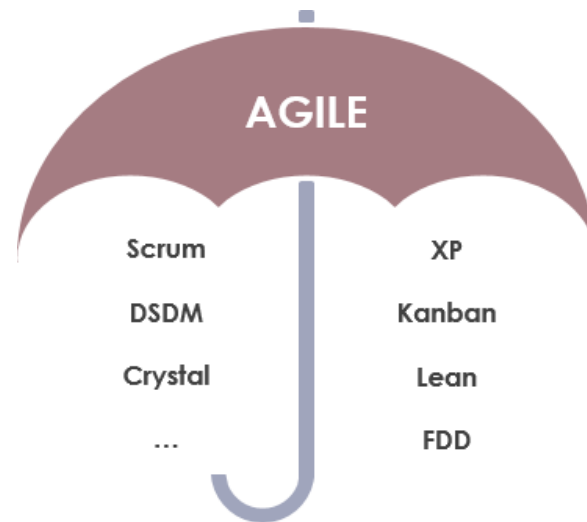
A powerful scrum software that supports scrum project management. It features scrum tools like user story map, product backlog management, sprint backlog management, task management, daily scrum meeting, sprint planning tool, sprint review tool, sprint retrospective tool, burndown, impediment, stakeholder and team management.

[Learn More \(/features/scrums-process-canvas/\)](/features/scrums-process-canvas/)



Scrum and other Leading Agile Methods

Agile is a mindset and it's a set of values and principles. Agile is a way of thinking and acting. Agile is all about short cycles, iterative and incremental delivery, failing fast, getting feedback, delivering business value to customers early and about people, collaboration and interaction. Agile is a mindset which is all about transparency, inspection and adaptation. Agile however doesn't consist of any [roles \(/scrum/what-are-the-three-scrum-roles/\)](#), events or [artifacts \(/scrum/what-are-scrum-artifacts/\)](#). It's a mindset. For example, [Scrum \(/scrum/scrum-in-3-minutes/\)](#) is one of the widely used frameworks under the Agile umbrella, which may help you in becoming more Agile, there are however many more frameworks within the Agile movement, like Kanban, XP, Crystal and many more as shown in the Figure below:



Scrum agile umbrella

Scrum

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It is used for managing software projects and product or application development. Its focus is on an adaptive product development strategy where a cross-functional team works as a unit to reach a common goal within 2-4 weeks ([Sprint \(/scrum/what-is-sprint-in-scrum/\)](#)). It consists of a collection of values, artifacts, roles, ceremonies, rules and best practices.

Lean



Lean originated with the Toyota Production System, or TPS, which revolutionized the manufacture of physical goods in the 1950s, '60s, and beyond. Lean maintains its hold in manufacturing but has also found new applications in knowledge work, helping businesses in all industries *eliminate waste, improve processes, and boost innovation*. Software development is a natural application of Lean methodology because, much like manufacturing, it generally follows a defined process, has some defined conditions of acceptance, and results in the delivery of tangible value. The key concepts that guide all practice of Lean methodology, which we call the Pillars of Lean. They are:

- Continuous improvement
- Respect for people
- Lightweight Leadership

Kanban

Kanban is a highly visual workflow management method that is popular among Lean teams. In fact, 83% of teams practicing Lean use Kanban to visualize and actively manage the creation of products with an emphasis on continual delivery, while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively.

Kanban is based on 3 basic principles:

- Visualize what you'll do today (workflow): Seeing all the items within the context of each other can be very informative
- Limit the amount of work in progress (WIP): This helps balance the flow-based approach so teams don't start and commit to too much work at once
- Enhance flow: When something is finished, the next highest priority item from the backlog is pulled into play

Kanban promotes continuous collaboration and encourages active, ongoing learning and improvement by defining the best possible team workflow.

Dynamic Systems Development Method (DSDM)



DSDM is a framework that is made up of eight principles, a lifecycle and products, roles and responsibilities and several best practice techniques. These underpin and support a philosophy of delivering strategically aligned business benefits as early as possible to give an organization the best possible return on investment (ROI).

DSDM is a methodology that prioritizes schedule and quality over functionality, which fixes cost, quality and time at the start and uses the MoSCoW method of prioritization, which breaks a project down into four different types of requirements:

- Must have (M)
- Should have (S)
- Could have (C)
- Won't have (W)

There are eight principles underpinning DSDM Atern[13]. These principles direct the team in the attitude they must take and the mindset they must adopt to deliver consistently.

1. Focus on the business need
2. Deliver on time
3. Collaborate
4. Never compromise quality
5. Build incrementally from firm foundations
6. Develop iteratively
7. Communicate continuously and clearly
8. Demonstrate control

Extreme Programming

Extreme Programming (XP), originally described by Kent Beck, has emerged as one of the most popular and controversial Agile methodologies. XP is a disciplined approach to delivering high-quality software quickly and continuously. It is intended to improve software quality and responsiveness in the face of changing customer



requirements. It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks.

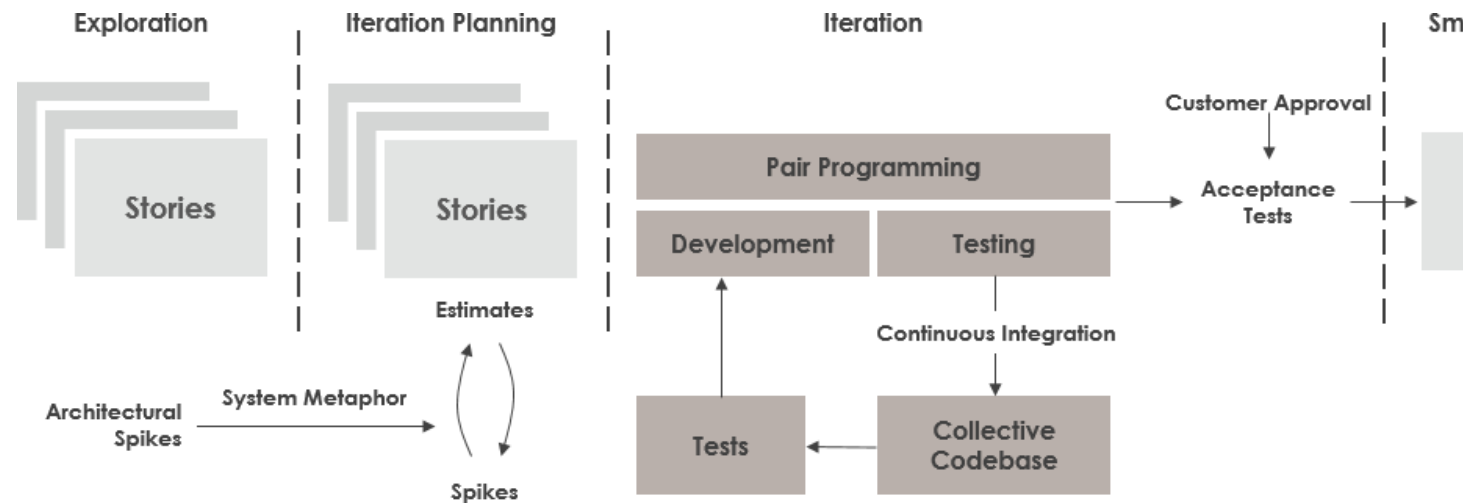
The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to “extreme” levels. As an example, code reviews are considered a beneficial practice. Taken to the extreme, code can be reviewed continuously through the practice of pair programming.

The original XP method is based on four simple values – simplicity, communication, feedback, and courage.

It also has twelve supporting practices:

- Planning Game
- Small Releases
- Customer Acceptance Tests
- Simple Design
- Pair Programming
- Test-Driven Development
- Refactoring
- Continuous Integration
- Collective Code Ownership
- Coding Standards
- Metaphor
- Sustainable Pace



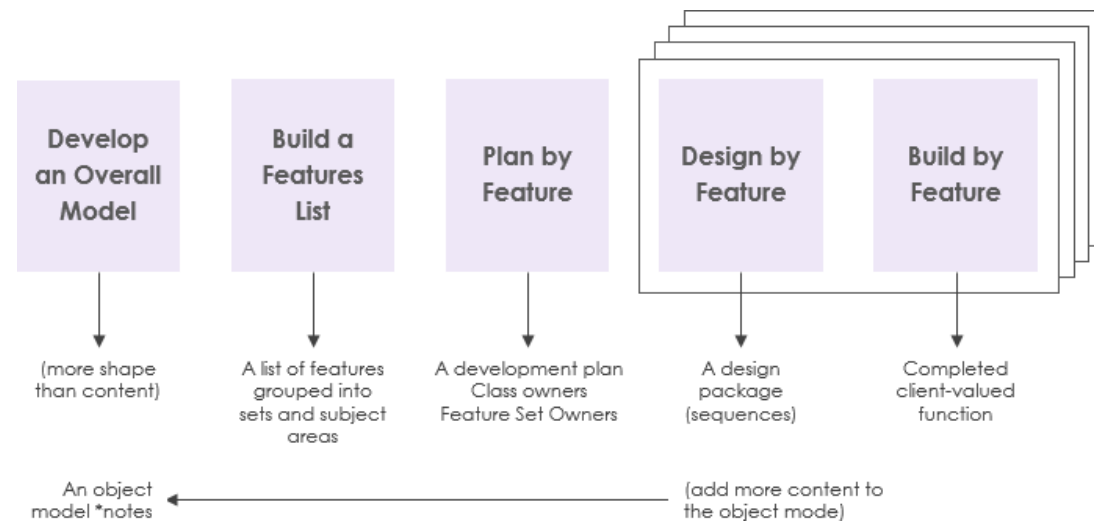


Extreme Programming

Feature Driven Development (FDD)

Feature-Driven Development (FDD) was introduced in 1997 by Jeff De Luca when he was working in a software development project for a large Singapore bank. It is an iterative and incremental software development process and is an agile method for developing software. FDD blends a number of industry-recognized best practices into a cohesive whole. These practices are driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner. The advantage of using FDD is that it is scalable even to large teams due to the concept of 'just enough design initially' (JEDI). It is a great solution to maintain control over agile, incremental and inherently complex projects because of its feature-centric process. It consists of five basic activities:

1. Development of an overall model
2. Building of a feature list
3. Planning by feature
4. Designing by feature
5. Building by feature.



Feature Driven Development (FDD)

Every project will have its own unique model, which will result in a feature list. The last three activities are short iterative processes, with a feature not taking longer than two weeks to build. If it will take more than two weeks, then it will have to be broken down into smaller features.

Crystal

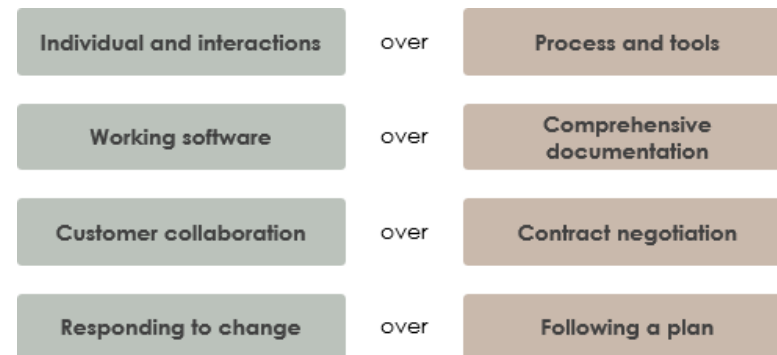
Crystal methods are a family of methodologies (the Crystal family) that were developed by Alistair Cockburn in the mid-1990s. The methods come from years of study and interviews of teams by Cockburn. Cockburn's research showed that the teams he interviewed did not follow the formal methodologies yet they still delivered successful projects. The Crystal family is Cockburn's way of cataloguing what they did that made the projects successful. Crystal methods are focused on:

- People
- Interaction
- Community
- Skills
- Talents
- Communications

Agile Manifesto



The term “Agile” was coined in 2001 in the Agile Manifesto. The manifesto set out to establish principles to guide a better approach to software development. The Agile Manifesto consists of 4 important values. The way to read the Agile Manifesto is not that the items on the right side have no value anymore, but the Agile movement values the items on the left more.



Agile manifesto

So let's take a look at the first line of the Agile Manifesto. This line states that we value people, their interactions, communication and collaboration more than having all sorts of extensive processes and tools in place. Of course processes and tools are valuable, however, they are much more valuable if they actually support people working together and delivering great products. What we see in a lot of organizations these days, is that the processes and tools are the goal themselves. From an Agile perspective, we value this differently. Processes and tools should be supporting people in working together and delivering value to customers.

Agile Manifesto Principles

Complementing the Agile Manifesto, the Agile Alliance has also defined a set of 12 underlying principles, which provide guidance and more detailed explanation in addition to the Agile Manifesto:





**Design with
the user**



**Build for
sustainability**



**Reuse and
improve**



**Understand
the existing
ecosystem**



**Be data
driven**



**Address
privacy &
security**



**Design for
scale**



**Use open
standards**



**Be
collaborative**

Agile Manifesto Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
10. Continuous attention to technical excellence and good design enhances agility.
11. Simplicity—the art of maximizing the amount of work not done—is essential.



12. The best architectures, requirements, and designs emerge from self-organizing teams. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Summary

Agile Development is one of the big buzzwords of the software development industry which is a different way of managing software development projects. Rather than a specific software development method, it is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.

