Agile Methodologies

The Agile methodologies outlined below share much of the same overarching philosophy, as well as many of the same characteristics and practices. From an implementation standpoint, however, each has its own unique mix of practices, terminology, and tactics.

The most widely-used Agile methodologies include:

- Agile Scrum Methodology
- Lean Software Development
- Kanban
- Extreme Programming (XP)
- Crystal
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD)

Agile Scrum Methodology

Scrum is a lightweight Agile project management framework that can be used to manage iterative and incremental projects of all types. It has gained increasing popularity over the years due to its simplicity, proven productivity, and ability to incorporate various overarching practices promoted by other Agile models.

In Scrum, the Product Owner works closely with their team to identify and prioritize system functionality by creating a Product Backlog. The Product Backlog consists of whatever needs to be done to up to deliver "potentially shippable increments" of software during successive Sprints, typically lasting 30 days. Once a Sprint's Product Backlog is committed, no additional functionality can be added to the Sprint except by the team. Once a Sprint has been delivered, the Product Backlog is analyzed and reprioritized, if necessary, and the next set of deliverables is selected for the next Sprint.

Lean Software Development

Lean Software Development is an iterative Agile methodology that focuses the team on delivering value to the customer through effective value stream mapping. It is a highly flexible, evolving methodology without rigid guidelines, rules, or methods.

The main principles of the Lean methodology include:

- Eliminating Waste
- Amplifying Learning
- Deciding as Late as Possible
- Delivering as Fast as Possible
- Empowering the Team
- Building Integrity In
- Seeing the Whole

Lean development eliminates waste by asking users to select only the truly valuable features for a system, prioritize those features, and then work to deliver them in small batches. It relies on rapid and reliable feedback between programmers and customers, emphasizing the speed and efficiency of development workflows. Lean uses the idea of a work product being "pulled" via customer request. It gives decision-making authority to individuals and small teams since this has been proven to be a faster and more efficient method than a hierarchical flow of control. Lean also concentrates on the efficient use of team resources, trying to ensure that everyone is as productive as possible for the maximum

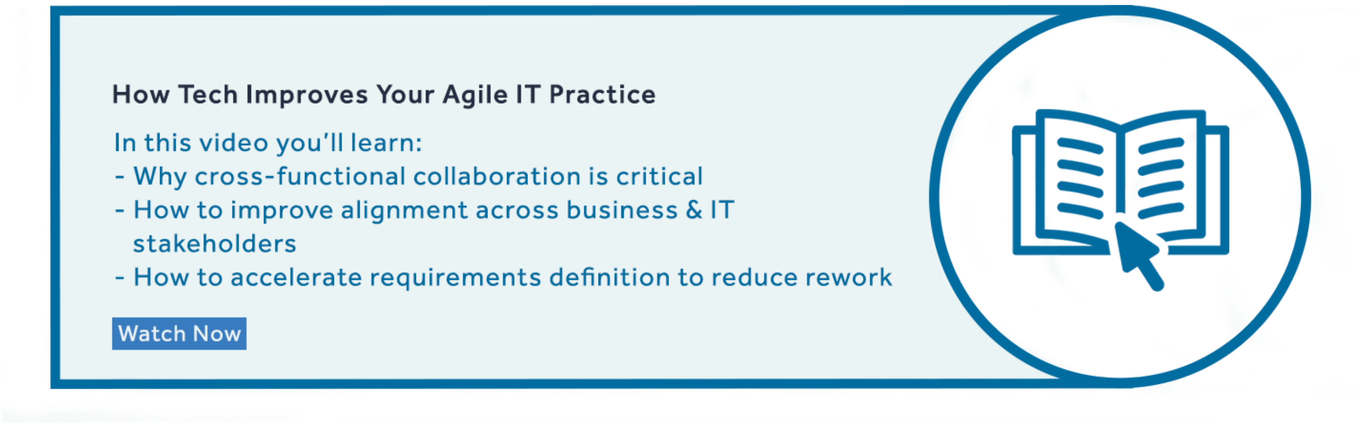
Kanban

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 adding more stress to the software development life cycle. 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 automation):*** 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.



How Tech Improves Your Agile IT Practice

In this video you'll learn:
- Why cross-functional collaboration is critical
- How to improve alignment across business & IT stakeholders
- How to accelerate requirements definition to reduce rework

Watch Now

Extreme Programming (XP)

Extreme Programming (XP), originally described by Kent Beck, has emerged as one of the most popular and controversial Agile models. XP is a disciplined approach for high-quality agile software development, focused on speed and continuous delivery. 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.

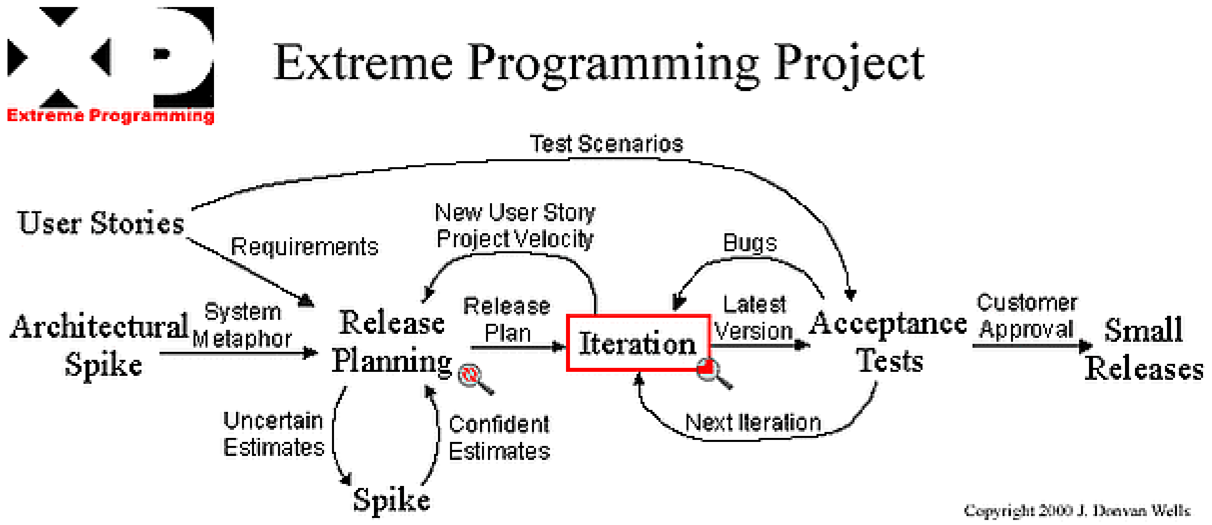
The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. As an example, code reviews are considered a

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

Don Wells has depicted the XP process in this popular diagram:



In XP, the customer works closely with the development team to define and prioritize user stories. The development team estimates, plans, and delivers the highest priority user stories in the form of working, tested software on an iteration-by-iteration basis. In order to maximize productivity, the practices provide a supportive, lightweight framework to guide a team and ensure high-quality enterprise software.

Crystal

Crystal is actually comprised of a family of Agile process models, including Crystal Clear, Crystal Yellow, Crystal Orange and others. Each has unique characteristics driven by several factors, such as team size, system criticality, and project priorities. This Crystal family addresses the realization that each project may require a slightly tailored set of policies, practices, and processes in order to meet the product 's unique characteristics.

Introduced by Alistair Cockburn, Crystal focuses primarily on people and the interaction among them while they work on an agile software development project. There is also a focus on business-criticality and business-priority of the system under development.

Unlike traditional development methods, Crystal doesn't try to fix the tools and techniques of development but keeps people and processes at the core of the process. However, it is not only the people or the processes that are important, rather the interaction between them that is most important.

Several key tenets of Crystal include teamwork, communication, and simplicity, as well as reflection to frequently adjust and improve the process. Like other Agile frameworks, Crystal promotes early, frequent delivery of working software, high user involvement, adaptability, and the removal of bureaucracy or distractions.

Dynamic Systems Development Method (DSDM)

The Dynamic Systems Development Method (DSDM) is an Agile approach that grew out of the need to provide a common industry framework for rapid software delivery. Since 1994, the DSDM methodology has evolved to provide a comprehensive foundation for planning, managing, executing, and scaling Agile process and iterative software development projects.

DSDM is based on eight key principles that direct the team and create a mindset to deliver on time and within budget. These agile principles primarily revolve around business needs/value, active user involvement, empowered teams, frequent delivery, integrated testing, and stakeholder collaboration. DSDM specifically calls out "fitness for business purpose" as the primary criteria for delivery and acceptance of a system, focusing on the useful 80% of the system that can be deployed in 20% of the time.

Compromising any of the following principles undermines the philosophy of DSDM and introduces risk to the successful outcome of the project.

DSDM's Eight Key Principles:

3. Collaborate
4. Never compromise quality
5. Build incrementally from firm foundations
6. Develop iteratively
7. Communicate continuously and clearly
8. Demonstrate control

Business Requirements are baselined at a high level early on in the project. Rework is built into the process, and all development changes must be reversible. System requirements are planned and delivered in short, fixed-length time-boxes – also known as sprints or iterations – and prioritized using MoSCoW Rules.

MoSCoW Rules:

M – Must have requirements
S – Should have if at all possible
C – Could have but not critical
W – Won't have this time, but potentially later

All critical work must be completed in a DSDM project's defined time-box. It is also important that not every requirement in a project or time-box is considered critical. Within each time-box, less critical items are also included so that they can be removed to keep from impacting higher priority requirements on the schedule.

Feature Driven Development (FDD)

Feature Driven Development is a model-driven, short-iteration process that was built around software engineering best practices such as domain object modeling, developing by feature, and code ownership. The blending of these practices that resulted in a cohesive whole is the best characteristic of FDD.

Feature Driven Development consists of five basic activities:

- Development of an overall model
- Building a feature list
- Planning by feature
- Designing by feature
- Building by feature

are small, “useful in the eyes of the client” results. If they will take more than two weeks to build, then they will have to be broken down into smaller features.

FDD’s main purpose is to deliver tangible, working software in a timely manner, repeatedly. The advantage of using FDD is that it is scalable even to large teams due to the concept of ‘just enough design initially’ (JEDI). Because of its feature-centric process, FDD is a great solution to maintain control for incremental and inherently complex Agile project management.

Gartner Report: 2019 Magic Quadrant for Enterprise Agile Planning Tools Get the Report ×

LEARNING RESOURCES

Why Storyteller

Resource Library

Events & Webinars

Blog

About Us

Partner Program

Careers

SIGN UP FOR OUR NEWSLETTER

Enter your email SIGN UP

90 Eglinton Ave E. #700,
Toronto, Ontario, M4P 2Y3 Canada