Continuous Improvement

Definition of Done

DevOps is a mindset, a culture, and a set of technical practices. It promotes communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, operate, and maintain a feature. A working agreement among the team defining the term “Done” (also called the Definition of Done or DOD) is crucial. This definition includes a checklist of items, with criteria for what is considered “Done.”

DevOps is the blending of the terms development and operations, rooted in an operational philosophy that promotes both communication and continuous high-quality software delivery and infrastructure deployment and maintenance.
DevOps 2.0

DevOps 2.0 is the extension of DevOps practices through the entire organization, beyond development and IT ops. DevOps 2.0 aims to break down silos and foster collaboration between developers, IT professionals, managers, and customers. It seeks to create a culture of continuous collaboration, market-driven transformation, and excellence, which will ultimately result in better products and services.

DevSecOps

DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps workflow from slowing down.

SecOps

SecOps is a management approach that connects security and operations teams, similar to how DevOps links software developers and operations professionals to increase efficiency and eliminate priority conflicts.

Epic

An Epic is a container for a technical development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval before implementation.

Feature

A Feature is a service that fills a key customer need. Each Feature includes a business hypothesis and acceptance criteria, and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a Program Increment (PI).

Framework

In computer systems, a framework is often a layered collection indicating what kind of programs can or should be built and how they should interrelate. Some computer system frameworks also include actual programs, specify programming interfaces, or offer programming tools for using the frameworks. A framework may be a set of functions within a system and the hierarchical relationships among those functions. Often, the communication should be standardized at some level of a network, and so forth, so a framework is generally more comprehensive than a protocol and more prescriptive than a specification.

Fail Fast

Fail fast is the process of starting work on a task or project, obtaining immediate feedback, and then determining whether to continue working on that task or take a different approach — that is, adapt. If a project is not working, it is best to determine that early on in the process rather than waiting until too much money and time has been invested.

Feedback

The key to successfully executing SAFe is to establish a continuous flow of work that supports incremental value delivery, based on constant feedback and adjustment. Continuous flow delivers feedback to the team, allowing for timely adjustments and improvements, and drives evidence-based governance.

Flow

Flow, in the context of psychology, is a state of intense engagement, focus, and concentration in the present moment. It occurs when a task is so compelling that one loses track of time and becomes absorbed in the activity. The experiences are considered profound enough to improve the individual's overall satisfaction in life.
Lean Software Development

Lean Software Development is an example of lightweight Agile methodology applied to project development. Lean Software Development combines the Lean manufacturing principles of just-in-time production and Lean IT principles, and applies them to software. It places a strong emphasis on people and effective communications.

Metrics

Metrics are agreed-upon measures used to evaluate how well the organization is progressing toward the products, large solution, program, and lead's business and technical objectives.

Minimal Viable Product

A Minimal Viable Product is, as Eric Ries said, "the "smallest" version of a product which allows a team to collect the maximum amount of validated learning about customers in the least possible time." The smallest viable thing you need to start validating the actual idea of the business or product. The very first increment of the business/product cycle (Agile Release Train) delivers the smallest viable product to the customers in a single PI. In Agile, a PI is a timeboxed increment of work during which the Agile Release Train builds and delivers an increment of value to the customers. 

Planning Poker

Planning Poker is a team building exercise or game used in Agile to arrive at a group consensus for estimating workload. It is a real-time estimation technique. Each team member estimates the hours required to complete a particular piece of work by putting down a number of chips (called "story points") on a table. The number of chips put down must be an integer value that is based on the team consensus. It is used by agile software development teams to perform relative estimation for a user story.

Platform

A platform is any hardware or software used to host an application or service. An application platform, for example, consists of programs, operating system, and middleware programs that make the instruction set for a particular processor or technology available. In this case, the platform creates a translation that ensures object code will execute successfully.

Product Owner

As a member of the Agile team, the Product Owner represents the customer, and conveys the customer's requirements and vision to the team. The product owner writes the acceptance criteria, and prioritizes and maintains the product backlog. The product owner also collaborates with the team to ensure that the team stays on track to meet the customer's vision for the product.

Program Backlog

A product backlog is a list of features changes to existing features, bug fixes, or other activities that a team may do in order to achieve a specific goal. The product backlog ensures that requirements are understood and prioritized with team concern to the customer and stakeholders, and ensuring that the team stays on track to meet the customer's vision for the product.

Relative Estimation

Relative estimation consists of estimating cards or user stories by comparison or by grouping of items of equivalent difficulty.

Release

A release is the distribution of the final version of an application. A software release may be either public or private, and may be delivered to users in the wild. A release is preceded by the distribution of alpha and beta versions of the software. A release can be made before the end of an iteration.
**Sprint**

Sprint is a process framework used to manage product development and other knowledge work.

...and the business needs it supports. A Sprint is a timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:

...and the business needs it supports. A Sprint is a timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:

**Sprint Meeting**

The Daily Scrum is one of the most commonly practiced Agile techniques and provides an opportunity for a Scrum Team to get together on a regular basis to coordinate their activities.

The Daily Scrum is one of the most commonly practiced Agile techniques and provides an opportunity for a Scrum Team to get together on a regular basis to coordinate their activities.

**Sprint Backlog**

A Sprint Backlog is the subset of a Product Backlog that the Scrum Team is committed to completing during a Sprint.

A Sprint Backlog is the subset of a Product Backlog that the Scrum Team is committed to completing during a Sprint.

**Sprint Goal**

A Sprint Goal is a high-level summary of the business and technical goals that the Scrum Team agrees to accomplish in an iteration. They are used in coordinating an Agile Release Train (ART) to align the work of functional groups, other ARTs, and external stakeholders.

A Sprint Goal is a high-level summary of the business and technical goals that the Scrum Team agrees to accomplish in an iteration. They are used in coordinating an Agile Release Train (ART) to align the work of functional groups, other ARTs, and external stakeholders.

**Sprint**

Sprint is the timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:

...and the business needs it supports. A Sprint is a timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:

**Scrum Meeting**

The Daily Scrum is one of the most commonly practiced Agile techniques and provides an opportunity for a Scrum Team to get together on a regular basis to coordinate their activities.

The Daily Scrum is one of the most commonly practiced Agile techniques and provides an opportunity for a Scrum Team to get together on a regular basis to coordinate their activities.

**Scrum**

Scrum is a process framework used to manage product development and other knowledge work.

...and the business needs it supports. A Sprint is a timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:

...and the business needs it supports. A Sprint is a timebox during which the Scrum Team produces a potentially shippable set of product increments. The boundaries of a Sprint are fixed—two to four weeks in duration, from project to project, and usually measured in one week. In addition, the Scrum Team commits to do the work that is in the Sprint Backlog. The boundaries of the Sprint are also fixed:
Agile Development Terms of Reference

SAFe®: Scaling Agile at Enterprise Scale

**Sprint Planning**

Sprint planning is an event that occurs at the **Beginning of a sprint** where the team identifies the **product backlog items** that they will work on during that sprint.

**Velocity**

Velocity is a useful planning tool for estimating how fast the development team can **productively** work on the product backlog items. It is calculated by reviewing the **product backlog items** that were completed during that sprint. The total is called **velocity**.

**Sprint Retrospective**

The Sprint Retrospective is a **regular** meeting where Agile Team members **review** the outcomes of the Iterations, **improve** their practices, and identify ways to **improve**.

**Sprint Review**

The Sprint Review is a customer-based event, where each team presents its work in **public**. It is **exclusive** to team members. The team brandishes its project to generate interest and to **build** the **market** for the next iteration.

**Story Point**

A story point is a metric used in agile project management and development to estimate the difficulty of implementing a story item. It is used in particular to assign value to software development tasks. Using estimations of story points rather than **man-hours** to determine the complexity of a story is a **complex** task. Story points are a useful way to understand if an item should be assigned more story points.

**Test-Driven Development**

Test-Driven Development (TDD) is an approach that involves writing tests before writing code, and then building code that will **pass** these tests.

**User Story**

An agile user story is a short description of a **functionality** or **value** that the users want to achieve. It is a **non-technical** description of the desired **behavior**. The term was first introduced in a paper published in 1970 by **Dijkstra** and has since been widely adopted in agile methodologies. A user story is a brief, non-technical description of a software system requirement written from the customer’s or end-user’s point of view.

**Velocity**

Velocity is a metric that predicts how much work a team can **complete** within a single sprint. It is calculated by reviewing the **product backlog items** that were completed during that sprint. The total is called **velocity**.

**User-End User**

The actual person operating the system in the field, or a **representation** of the end user.

**Waterfall**

Similar to the direction water flows over the edge of a cliff, distinct endpoints or **goals** are set for each phase of the software development lifecycle. Each phase must be completed before moving on to the next. This model was developed by **Royce** and is still widely used today.

**Waterfall Model**

Waterfall Model is a linear approach to software development. It emphasizes upfront planning and sequential development phases. The product is developed in stages, with each stage building upon the previous one. This approach is often used in large-scale projects where the scope and requirements are well-defined from the start. However, it can be challenging if changes are needed later in the process.

Each sprint begins with a planning meeting. During the meeting, the team discusses the **product backlog items** and the development team aims to complete as many of these items as possible during that sprint. The team determines the length of the sprint. The sprint planning meeting is **immediate** following the completion of a sprint and identifies actions for improvement going forward.

A Scrum retrospective is a meeting held at the end of a **Sprint** to discuss what happened during that Sprint and identify areas for improvement going forward. Agile teams typically hold a sprint retrospective after every sprint to review and discuss what went well and what could be improved in the next sprint.

A Scrum retrospective is an opportunity for the Scrum team to reflect on their **performance** and discuss what went well and what could be improved in the next sprint. It is held at the conclusion of each sprint and typically lasts about two hours. The retrospective is an important part of the Scrum process and helps ensure that the team continues to improve and deliver value to its stakeholders.