



A

Acceptance Test Driven Development (ATDD)

Acceptance Test Driven Development (ATDD) involves team members with different perspectives (customer, development, testing) collaborating to write acceptance tests in advance of implementing the corresponding functionality.

Acceptance Testing

An acceptance test is a formal description of the behavior of a software product, generally expressed as an example or a usage scenario. A number of different notations and approaches have been proposed for such examples or scenarios. In many cases the aim is that it should be possible to automate the execution of such tests by a software tool, either ad-hoc to the development team or off the shelf.

Antipattern

Antipatterns are common solutions to common problems where the solution is ineffective and may result in undesired consequences.

Automated Build

In the context of software development, build refers to the process that converts files and other assets under the developers' responsibility into a software product in its final or consumable form. The build is automated when these steps are repeatable, require no direct human intervention, and can be performed at any time with no information other than what is stored in the source code control repository.

B

Backlog

See [product backlog](#).

Backlog Grooming / Backlog Refinement

Backlog grooming is when the product owner and some, or all, of the rest of the team refine the backlog on a regular basis to ensure the backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery.

Behavior Driven Development (BDD)

BDD is a practice where members of the team discuss the expected behavior of a system in order to build a shared understanding of expected functionality.

Burndown Chart

Burndown charts and burnup charts track the amount of output (in terms of hours, story points, or backlog items) a team has completed across an iteration or a project.

Business Agility

Business agility is the ability of an organization to sense changes internally or externally and respond accordingly in order to deliver value to its customers.

C

Collective Ownership

Collective code ownership is the explicit convention that every team member can make changes to any code file as necessary: either to complete a development task, to repair a defect, or to improve the code's overall structure.

Continuous Deployment

Continuous deployment aims to reduce the time elapsed between writing a line of code and making that code available to users in production. To achieve continuous deployment, the team relies on infrastructure that automates and instruments the various steps leading up to deployment, so that after each integration successfully meeting these release criteria, the live application is updated with new code.

Continuous Integration

Continuous Integration is the practice of merging code changes into a shared repository several times a day in order to release a product version at any moment. This requires an integration procedure which is reproducible and automated.

CRC Cards

Class Responsibility Collaborator (CRC) Cards are an object oriented design technique teams can use to discuss what a class should know and do and what other classes it interacts with.

Customer Development

Customer development is a four-step framework that provides a way to use a scientific approach to validate assumptions about your product and business. ([learn more](#))

D

Daily Meeting

The daily meeting is one of the most commonly practiced Agile techniques and presents opportunity for a team to get together on a regular basis to coordinate their activities.

Definition of Done

The definition of done is an agreed upon list of the activities deemed necessary to get a product increment, usually represented by a user story, to a done state by the end of a sprint.

Definition of Ready

Definition of Ready involves creating clear criteria that a user story must meet before being accepted into an upcoming iteration. This is typically based on the INVEST matrix.

E

Epic

An epic is a large user story.

Estimation

In software development, an “estimate” is the evaluation of the effort necessary to carry out a given development task; this is most often expressed in terms of duration.

Exploratory Testing

Exploratory testing is, more than strictly speaking a “practice,” a style or approach to testing software which is often contrasted to “scripted testing.”

Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

F

Facilitation

A facilitator is a person who chooses or is given the explicit role of conducting a meeting.

Frequent Releases

An Agile team frequently releases its product into the hands of end users, listening to feedback, whether critical or appreciative.

G

Given When Then

The Given-When-Then formula is a template intended to guide the writing of acceptance tests for a User Story: (Given) some context, (When) some action is carried out, (Then) a particular set of observable consequences should obtain.

I

Incremental Development

In an Agile context, Incremental Development is when each successive version of a product is usable, and each builds upon the previous version by adding user-visible functionality.

Information Radiators

“Information radiator” is the term for any of a number of visual displays which a team places in a highly visible location, so that all team members can see the latest information at a glance.

Integration

“Integration” (or “integrating”) refers to any efforts still required for a project team to deliver a product suitable for release as a functional whole.

INVEST

The acronym INVEST stands for a set of criteria used to assess the quality of a user story. If the story fails to meet one of these criteria, the team may want to reword it.

Iteration

An iteration is a timebox during which development takes place. The duration may vary from project to project and is usually fixed.

Iterative Development

Agile projects are iterative insofar as they intentionally allow for “repeating” software development activities, and for potentially “revisiting” the same work products (the phrase “planned rework” is sometimes used; refactoring is a good example).

K

Kanban

The Kanban Method is a means to design, manage and improve flow for knowledge work and allows teams to start where they are to drive evolutionary change.

Kanban Board

A Kanban Board is a visual workflow tool consisting of multiple columns. Each column represents a different stage in the workflow process.

L

Lead Time

Lead Time is the time between a customer order and delivery. In software development, it can also be the time between a requirement made and its fulfillment.

M

Milestone Retrospective

A Milestone Retrospective is a team's detailed analysis of the project's significant events after a set period of time or at the project's end.

Minimum Marketable Feature (MMF)

A Minimum Marketable Feature is a small, self-contained feature that can be developed quickly and that delivers significant value to the user.

Minimum Viable Product (MVP)

A Minimum Viable Product is, as Eric Ries said, the “version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.”

Mob Programming

Mob Programming is a software development approach where the whole team works on the same thing, at the same time, in the same space, and at the same computer.

Mock Objects

Mock Objects (commonly used in the context of crafting automated unit tests) consist of instantiating a test-specific version of a software component.

N

Niko-niko Calendar

A Niko-niko Calendar is updated daily with each team member's mood for that day. Over time the calendar reveals patterns of change in the moods of the team, or of individual members.

O

Open Space

In Open Space meetings, events, or conferences, participants create and manage their own agenda of parallel sessions around a specific theme.

P

Pair Programming

Pair programming consists of two programmers sharing a single workstation (one screen, keyboard and mouse among the pair).

Personas

Personas are synthetic biographies of fictitious users of the future product.

Planning Poker

An approach to estimation used by Agile teams. Each team member "plays" a card bearing a numerical value corresponding to a point estimation for a user story.

Points (estimates in)

Agile teams generally prefer to express estimates in units other than the time-honored "man-hours." Possibly the most widespread unit is "story points."

Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

Product Owner

The product owner is a role created by the Scrum Framework responsible for making sure the team delivers the desired outcome.

Project Chartering

A high-level summary of the project's key success factors displayed on one wall of the team room as a flipchart-sized sheet of paper.

Q

Quick Design Session

When “simple design” choices have far-reaching consequences, two or more developers meet for a quick design session at a whiteboard.

R

Refactoring

Refactoring consists of improving the internal structure of an existing program's source code, while preserving its external behavior.

Relative Estimation

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

Retrospective

The team meets regularly to reflect on the most significant events that occurred since the previous such meeting, and identify opportunities for improvement.

Rule of Simplicity

Rules of Simplicity is a set of criteria, in priority order, proposed by Kent Beck to judge whether some source code is “simple enough.”

S

Scrum

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

Scrumban

Scrumban is a mixture of [Scrum](#) and [Kanban](#).

For a more in-depth look at Scrumban, [please see this post](#) written by the creator of the Scrumban Method, Corey Ladas.

Scrum Master

The scrum master is responsible for ensuring the team lives agile values and principles and follows the practices that the team agreed they would use.

Scrum of Scrums

A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10.

Sign Up for Tasks

Members of an Agile development team normally choose which tasks to work on, rather than being assigned work by a manager.

Simple Design

A team adopting the “simple design” practice bases its software design strategy on a set of “simple design” principles.

Sprint Backlog

A sprint backlog is the subset of product backlog that a team targets to deliver during a sprint in order to accomplish the sprint goal and make progress toward a desired outcome.

Sprint Planning

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

Story Mapping

Story mapping consists of ordering user stories along two independent dimensions.

Story Splitting

Splitting consists of breaking up one user story into smaller ones, while preserving the property that each user story separately has measurable business value.

Sustainable Pace

The team aims for a work pace that they would be able to sustain indefinitely.

Spikes

are used to investigate a concept and/or create a simple prototype. Basically, a spike is a quick (typically less than a sprint of effort) exploration by coding of an area in which the development team lacks confidence

T

Task Board

The most basic form of a task board is divided into three columns labeled “To Do,” “In Progress,” and “Done.” Cards are placed in the columns to reflect the current status of that task.

Test Driven Development (TDD)

“Test-driven development” is a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring).

Team

A “team” in the Agile sense is a small group of people, assigned to the same project or effort, nearly all of them on a full-time basis.

Team Room

The team (ideally the whole team, including the product owner or domain expert) has the use of a dedicated space for the duration of the project, set apart from other groups’ activities.

Three C’s

“Card, Conversation, Confirmation” is a formula that captures the components of a User Story.

Three Amigos

Three amigos refers to the primary perspectives to examine an increment of work before, during, and after development. Those perspectives are Business, Development, and Testing.

Three Questions

The daily meeting is structured around some variant of the following three questions: What have you completed? What will you do next? What is getting in your way?

Timebox

A timebox is a previously agreed period of time during which a person or a team works steadily towards completion of some goal.

U

Ubiquitous Language

Striving to use the vocabulary of a given business domain, not only in discussions about the requirements for a software product, but in discussions of design as well and all the way into “the product’s source code itself.”

Unit Testing

A unit test is a short program fragment written and maintained by the developers on the product team, which exercises some narrow part of the product’s source code and checks the results.

Usability Testing

Usability testing is an empirical, exploratory technique to answer questions such as “how would an end user respond to our software under realistic conditions?”

User Stories

In consultation with the customer or product owner, the team divides up the work to be done into functional increments called “user stories.”

User Story Template

The user story template is one of the most commonly recommended aids to write user stories: As a ... I want ... So that ...

V

Velocity

At the end of each iteration, the team adds up effort estimates associated with user stories that were completed during that iteration. This total is called velocity.

Version Control

Version control is not strictly an Agile “practice” insofar as it is now widespread in the industry as a whole. But it is mentioned here for several reasons.

X

XP – Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.