**Acceptance Criteria1:**

The conditions that a software product must satisfy to be accepted by a user, customer, or in the case of system level functionality, the consuming system.

Acceptance Criteria are a set of statements, each with a clear pass/fail result, that specify both functional and non-functional requirements, and are applicable at the Epic, Feature, and Story Level. Acceptance criteria constitute our “Definition of Done”, and by done I mean well done.

We’re not talking about horseshoes here, and there is no partial acceptance: either the acceptance criteria is met or it is not.

**Backlog2:**

A backlog is a list of features or technical tasks which the team maintains and which, at a given moment, are known to be necessary and sufficient to complete a project or a release:

* if an item on the backlog does not contribute to the project's goal, it should be removed;
* on the other hand, if at any time a task or feature becomes known that is considered necessary to the project, it should be added to the backlog.

These "necessary and sufficient" properties are assessed relative to the team's state of knowledge at a particular moment; the backlog is expected to change throughout the project's duration as the team gains knowledge.

The backlog is the primary point of entry for knowledge about requirements, and the single authoritative source defining the work to be done.

**Backlog Grooming2:**

Backlog grooming is when the product owner and some, or all, of the rest of the team review items on the backlog 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. This activity occurs on a regular basis and may be an officially scheduled meeting or an ongoing activity. Some of the activities that occur during this refinement of the backlog include:

* removing user stories that no longer appear relevant
* creating new user stories in response to newly discovered needs
* re-assessing the relative priority of stories
* assigning estimates to stories which have yet to receive one
* correcting estimates in light of newly discovered information
* splitting user stories which are high priority but too coarse grained to fit in an upcoming iteration

**Definition of Done2:**

The team agrees on, and displays prominently somewhere in the team room, a list of criteria which must be met before a product increment "often a user story" is considered "done". Failure to meet these criteria at the end of a sprint normally implies that the work should not be counted toward that sprint's velocity.

**Epic2:**

An epic is a large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.

There is no standard form to represent epics. Some teams use the familiar user story formats (As A, I want, So That or In Order To, As A, I want) while other teams represent the epics with a short phrase.

**Minimum Viable Product (MVP) 2:**

A minimum viable product (MVP) is a concept from Lean Startup that stresses the impact of learning in new product development. Eric Ries, defined an MVP as that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort. This validated learning comes in the form of whether your customers will actually purchase your product.

A key premise behind the idea of MVP is that you produce an actual product (which may be no more than a landing page, or a service with an appearance of automation, but which is fully manual behind the scenes) that you can offer to customers and observe their actual behavior with the product or service. Seeing what people actually do with respect to a product is much more reliable than asking people what they would do.

**Sprint or Iteration2:**

An iteration, in the context of an Agile project, is a timebox during which development takes place, the duration of which:

* may vary from project to project, usually between 1 and 4 weeks
* is in most cases fixed for the duration of a given project

A key feature of Agile approaches is the underlying assumption that a project consists exclusively of a sequence of iterations, possibly with the exception of a very brief "vision and planning" phase prior to development, and a similarly brief "closure" phase after it.

In general iterations are aligned with calendar weeks, often starting on Mondays and ending on Fridays; this is more a matter of convenience than an explicit recommendation and many teams adopt different conventions.

The fixed length of iterations gives teams a simple way to obtain, based on velocity and the amount of work remaining, a usually accurate (though not very precise) estimation of the project's remaining duration.

**Stand-up/daily meeting2:**

Each day at the same time, the team meets so as to bring everyone up to date on the information that is vital for coordination: each team members briefly describes any "completed" contributions and any obstacles that stand in their way. Usually, Scrum's Three Questions are used to structure discussion. The meeting is normally held in front of the task board.

This meeting is normally timeboxed to a maximum duration of 15 minutes, though this may need adjusting for larger teams. To keep the meeting short, any topic that starts a discussion is cut short, added to a "parking lot" list, and discussed in greater depth after the meeting, between the people affected by the issue.

**Task/Story Board2:**

In its most basic form, a task board can be drawn on a whiteboard or even a section of wall. Using electrical tape or a dry erase pen, the board is divided into three columns labeled "To Do", "In Progress" and "Done". Sticky notes or index cards, one for each task the team is working on, are placed in the columns reflecting the current status of the tasks.

Many variants exist. Different layouts can be used, for instance by rows instead of columns (although the latter is much more common). The number and headings of the columns can vary, further columns are often used for instance to represent an activity, such as "In Test".

The task board is updated frequently, most commonly during the daily meeting, based on the team's progress since the last update. The board is commonly "reset" at the beginning of each iteration to reflect the iteration plan.

**User Stories2:**

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

Each user story is expected to yield, once implemented, a contribution to the value of the overall product, irrespective of the order of implementation; these and other assumptions as to the nature of user stories are captured by the INVEST formula.

To make these assumptions tangible, user stories are reified into a physical form: an index card or sticky note, on which a brief descriptive sentence is written to serve as a reminder of its value. This emphasizes the "atomic" nature of user stories and encourages direct physical manipulation: for instance, decisions about scheduling are made by physically moving around these "story cards".

Sources:

1) <https://www.leadingagile.com/2014/09/acceptance-criteria/>

2) <https://www.agilealliance.org/agile101/agile-glossary/>