The Agile Manifesto is based on 12 principles:

- Customer satisfaction by rapid delivery of a useful product solution
- Welcome changing requirements, even late in development
- Working products are delivered frequently (weeks rather than months)
- Close, daily cooperation between business people and developers
- Projects are built around motivated individuals, who should be trusted
- Face-to-face conversation is the best form of communication (co-location)
- Working solutions is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity, the art of maximizing the amount of work that is not done is essential
- Self-organizing teams
- Regular adaptation to changing circumstance

1. Which of the following best describes the approach for determining the iteration (timebox) length?
   a. Iterations (timeboxes) should always be 30 days
   b. The team determines iteration (timebox) length by dividing the total number of story points by the average velocity of the team
   c. Iterations (timeboxes) should always be two weeks
   d. The team should agree on the length of the iteration (timebox), taking the size and complexity of the project into consideration

2. Which of the following is a characteristic of an Agile leader?
   a) Task focused
   b) Process oriented
   c) Supportive
   d) Disengaged
3. Who is responsible for prioritizing the product backlog?
   a) Product Owner
   b) Project Manager
   c) Lead Developer
   d) Business Analyst

4. What are the advantages of maintaining consistent iteration (timebox) length throughout the project?
   a) It helps to establish a consistent pattern of delivery
   b) It helps the team to objectively measure progress
   c) It provides a consistent means of measuring team velocity
   d) All of the above

5. Tracking project issues in an Agile project is the primary responsibility of the…
   a) Tester
   b) Project Leader
   c) Functional Manager
   d) Developer

6. Why is it important to trust the team?
   a) High trust teams do not have to be accountable to each other
   b) High trust teams do not require a user representative
   c) The Project Manager does not then have to keep a project schedule
   d) The presence of trust is positively correlated with the team performance

7. An effective workshop facilitator will always …
   a) Involve the whole project team in all project workshops
   b) Agree the process and participants of the workshop with the workshop owner before the workshop
   c) Involve only those team members who will commit to doing further work after the workshop
   d) Act as a proxy for any invited participant who is unable to attend the workshop on the day

8. Which of the following best represents the Agile approach to planning?
   a) Planning is not part of an Agile approach, because Agile is exploratory
   b) Planning should be done in detail at the outset of a project and not revisited
   c) Planning should involve the whole team, not just the Project Manager
   d) Planning should all be done by the Project Manager

9. Who should define the business value of a Feature within an Agile project?
   a. The individual end-users
   b. The Product Owner
   c. The Business Analyst
   d. The Business Sponsor
10. If a timebox (iteration) plan needs to be reprioritized in a hurry, who should re-prioritize?
   a. The developers alone (they know what the customer wants)
   b. The Product Owner (the developers would only choose the easy things as top priority)
   c. The Project Leader (they can give an independent, pragmatic view)
   d. The whole team including Product Owner and developers (together they can consider both business value and practicality)

1. A user story is finished when:
   a) The conditions of satisfaction are achieved
   b) Design, coding, unit testing and testing tasks of this story are finished
   c) The team Done definition criteria are satisfied
   d) The Product Owner reviews it in the iteration review meeting

2. The product owner is responsible of which of the following activities:
   1. Represent the Customer/users community
   2. Prioritize features according to business value
   3. Ensure that the team is fully functional and productive
   4. Make scope/schedule tradeoff decisions
   5. Protects the team from external conflicts and interferences
   Answers:
   a) 1, 2, 3, 4
   b) 1, 2, 4
   c) 2, 4, 5
   d) All the above

3. The implementation details of a user story are discovered:
   a) During the story writing workshops with the Product Owner
   b) During the iteration review meeting with Product Owner
   c) As the team progresses towards the end of the release
   d) During the iteration where this user story is developed

4. One of the basic benefits on using continuous integration is:
   a) The elimination of integration bugs
   b) Ensuring release is free of bugs
   c) Early detection of integration problems
   d) Ensuring baselines are correctly taken

5. In Agile software development, the best measure of progress is:
   a) Percentage of resolved bugs
   b) Working software
   c) Phase completion
   d) Release burn charts
6. The primary role of the Scrum Master in Planning Poker estimation meeting is to:

   a) Facilitate the discussions among the team members and make sure they are following the steps of planning poker properly
   b) Participate in the estimation and challenge the team estimates to make sure they are estimating correctly
   c) Make sure that everyone is estimating by himself/herself without being influenced by others
   d) Make sure that the Product Owner estimates don't influence the team estimates

7. The purpose of the daily standup meeting is:

   a) Track the remaining effort of the daily team tasks
   b) Analyze the daily issues as soon as they emerge
   c) **Track the team's daily progress and highlight any pending issues**
   d) **Track the team's daily progress and solve any pending issues**

8. An iteration is over when:

   a) The iteration tasks are completed
   b) The stories are completed according to the Done definition
   c) **The timebox expires**
   d) The iteration review meeting and retrospective are conducted

• **Acceptance criteria**

These are specific criteria identified by the customer for each functional requirement. The acceptance criteria are written in simple terms and from a perspective of the customer.

Sample format is:

As a.....I want to..... So that I can.....
• **Acceptance testing**

Acceptance testing is a validation activity conducted to determine whether or not a system satisfies its acceptance criteria. It is the last phase of the product testing process.

• **Agile**

A conceptual framework for creating and then integrating a collection of partial solutions to produce a successful project. Agile methods are a family of development processes, not a single approach to project development.

• **Behavior Driven Development**

Behavior driven development (or BDD) is an agile product development technique that encourages collaboration between developers, QA and non-technical or business participants in a software project. BDD focuses on obtaining a clear understanding of desired software behavior through discussion with stakeholders. It extends TDD by writing test cases in a natural language that non-technical users can read.

• **Bottleneck**

A bottleneck is a sort of congestion in a system that occurs when workload arrives at a given point more quickly than that point can handle it.

It is metaphorically derived from the flowing of water through a narrow mouthed bottle where the flow of water is constrained by the size of its neck.

• **Bugs**

A bug is a problem causing a proposed solution to fail or produce invalid results. It is caused by insufficient or erroneous logic and can be an error, mistake, defect or fault.

• **Burndown Chart**

A burndown chart is a visual tool for measuring and displaying progress. Visually, a burndown chart is simply a line chart representing remaining work over time. Burndown charts are used to measure the progress of an agile project at both an iteration and project level.

• **Daily Standup/Scrum**

A Daily Standup is a whole team meeting that happens at the same time every day that usually lasts 15 minutes or less. The meeting is designed to allow the entire team to
synchronize with each other and to understand the flow and challenges of the development process. Each team member should provide the following information:
what did I do yesterday, what am I planning to do today, and what impediments do I currently have?

- **Done**

  Also referred to as “Done Done”, this term is used to describe all the various tasks that need to happen before a story is considered potentially releasable.

- **Epic**

  A very large user story that is eventually broken down into smaller stories.

- **Estimation**

  The process of agreeing on a size measurement for the stories, as well as the tasks required to implement those stories, in a product backlog.

- **Feature creep**

  Feature creep occurs when a proposed solution becomes overly complicated and difficult to use as a result of too many features.

- **Kanban**

  Kanban, pronounced /ˈkɑnˈbɑn/, is a method for developing products with an emphasis on just-in-time delivery and the optimization of flow of work on the team. It emphasizes that developers pull work from a queue, and the process, from definition of a task to its delivery to the customer, is displayed for participants to see.

- **Lean**

  Lean project development is a translation of Lean manufacturing and Lean IT principles and practices to the software development domain. Adapted from the Toyota Production System and is a set of techniques and principles for delivering more values with the same or less resources by eliminating waste across organizations and business processes.

- **Pair Programming/Pair Production**

  Pair production is an agile project development technique in which two team members work together. One defines a solution while the other reviews each line of the definition.
as it is put forward. The person defining the solution is called the driver. The person reviewing the developing solution is called the observer (or navigator). The two team members switch roles frequently.

- **Planning Poker**

  Also called Scrum poker, is a consensus-based technique for estimating, mostly used to estimate effort or relative size of tasks in project development.

- **Product Backlog**

  Acts as a repository for requirements targeted for release at some point. These are typically high level requirements with high level estimates provided by the product stakeholders. The requirements are listed on the backlog in priority order and maintained by the product owner.

- **Product Owner**

  The Product Owner represents the voice of the customer and is accountable for ensuring that the Team delivers value to the business. The Product Owner writes customer-centric items (typically user stories), prioritizes them, and adds them to the product backlog. Scrum teams should have one Product Owner.

- **Retrospective**

  A team meeting that happens at the end of every development iteration to review lessons learned and to discuss how the team can be more efficient in the future. It is based on the principles of applying the learning from the previous sprint to the upcoming sprint.

- **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 based on the adaptive and iterative methodology of project development.

- **Scrumban**

  Scrumban is a mix between Scrum and Kanban, which supposedly contains the best features of both methods.
- **Scrum Master**

Scrum is accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables. The ScrumMaster is not the team leader but acts as a buffer between the team and any distracting influences. The ScrumMaster ensures that the Scrum process is used as intended. The ScrumMaster is the enforcer of rules. A key part of the ScrumMaster's role is to protect the team and keep them focused on the tasks at hand. The role has also been referred to as servant-leader to reinforce these dual perspectives.

- **Spike**

A short, time-boxed piece of research, usually technical, on a single story that is intended to provide just enough information that the team can estimate the size of the story.

- **Sprint / Iteration**

A fixed duration period of time where user stories are chosen to work on. The term Sprint comes from the Scrum methodology and is analogous to the term Iteration. A sprint is defined as a 2-4 week increment of development activities that delivers working solutions at the end of the increment. External influences are not allowed to change the requirements of the stories being worked on.

- **Sprint Backlog**

At the beginning of each sprint, the team has sprint planning with an end result being a backlog of work that the team anticipates completing at the end of the sprint. These are the items that the team will deliver against throughout the duration of the sprint.

- **Sprint Planning**

Is a pre-sprint planning meeting attended by the core agile team? During the meeting the Product Owner describes the highest priority features to the team as described on the product backlog. The team then agrees on the number of features they can accomplish in the sprint and plans out the tasks required to achieve delivery of those features. The planning group works the features into User Stories and assigns Acceptance criteria to each story.

- **Sprint Review**

Each Sprint is followed by a Sprint review. During this review the solutions developed in the previous Sprint are reviewed and if necessary new backlog items are added.
• **Story Points**

  Unit of estimation measuring complexity and size.

• **Task**

  A user story can be broken down into one or more tasks. Tasks are estimated daily in hours (or story points) remaining by the developer working on them.

• **Taskboard / Storyboard**

  A wall chart with cards and sticky notes that represents all the work for in a given sprint. The notes are moved across the board to show progress.

• **Team**

  The Team is responsible for delivering the product. A Team is typically made up of 5–9 people with cross-functional skills who do the actual work (analyze, design, develop, test, technical communication, document, etc.). It is recommended that the Team be self-organizing and self-led, but often work with some form of project or team management.

• **Test Driven Development**

  Test-driven development (TDD) is a development process that relies on the repetition of a very short development cycle: first the developer writes a failing automated test case that defines a desired improvement or new function, then produces solutions to pass that test and finally refactors the new strategy to acceptable standards.

• **Timeboxing**

  Timeboxing is a planning technique common in planning projects (typically for software development), where the schedule is divided into a number of separate time periods (timeboxes, normally two to six weeks long), with each part having its own deliverables, deadline and budget.

• **User Persona**

  Personas are a description of the typical users of given software. A persona description should include:
  1) Skills and background – E.g. professional or beginner computer user
  2) Goals – E.g. what does the user expect from the product?
• **User Story**

A user story is a very high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it. A user story is one or more sentences in the everyday or business language of the end user that captures what the user wants to achieve. A user story is also a placeholder for conversation between the users and the team. The user stories should be written by or for the customers for a project and are their main instrument to influence the development of the solution. User stories could also be written by developers to express non-functional requirements (security, performance, quality, etc.)

• **Velocity**

It is a relative number which describes how much work the team can get done over a period of time.

• **Vertical Slice**

Showing off a feature in an application that works from start to finish but may be limited in scope. For example a rope bridge crossing a chasm is immediately useful and allows people to cross. Having that in place can help to build a better bridge later.

• **WIP**

Also known as Work in Progress is any work that has been started but has yet to be completed.

• **XP**

A software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles (timeboxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted.