

The 12 Agile Principles are a set of guiding concepts that support project teams in implementing agile projects. Use these concepts to implement agile methodologies in your projects.

For each of the sub-categories, provide the following weighted response:

Almost always or always = 5

Around 75% of the time = 4

About 50% of the time = 3

About 25% of the time = 2

Around 10% of the time = 1

Never = 0

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - a. User Story Cycle Time (time to complete a new User Story) averages 3 days or less.
  - b. Team understands and is efficiently managing their Work in Progress (WIP)
  - c. We deliver what we planned to in Sprint Planning (within a few story points)
    - >95% = 5
    - 90-95% = 4
    - 85-89% = 3
    - 75-84% = 2
    - 60-74% = 1
    - <60% = 0
  
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - a. Our team is quickly able to adjust to changes in the Product Backlog (requirements) and deliver working software. For example, adjusting on the day of Sprint Planning.
  - b. All code changes are reversible. We develop cohesive, loosely-coupled software.
  - c. We hold efficient, time-boxed Sprint Planning and Product Backlog Refinement (Grooming) sessions attended by an engaged Product Owner.
  
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - a. Our software baseline is in releasable state at the end of every sprint.  
*(this includes exploratory systems and analytical (but not formal) V&V)*
  - b. We spend minimal time on up-front design and documentation. We explore and learn by writing code.
  - c. Functional, non-functional, and integration testing is integrated into an automated or continuous build.

4. Business people and developers must work together daily throughout the project.
  - a. We have a Product Owner that works with the team daily to clarify requirements and verify that acceptance criteria are met and stories are done.
  - b. Our daily stand up lasts 15 minutes or less and is valuable to the team.
  - c. All work performed is captured in the Sprint Backlog
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
  - a. Our team is enthusiastic and has the right mix of skills and resources to successfully achieve our sprint goals.
  - b. Impediments are identified and removed rapidly by a dedicated team member.
  - c. Our team respects and trusts each other and identifies as a team.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - a. Software developers and testers collaborate or work in pairs to ensure quality before and after code is written.
  - b. User Stories are refined during face-to-face collaboration between the team and Product Owner / users.
  - c. Software is developed in pairs or is peer code reviewed face-to-face.
7. Working software is the primary measure of progress.
  - a. The User Stories we work on in the current sprint deliver demonstrable business value at the end of every sprint.
  - b. We have very few defects in our production software.
  - c. We have little or no technical debt in our product.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - a. User Stories are completed within the sprint without splitting or rolling over.
  - b. Code is written using a *test-code-refactor* (TDD) approach or with aggressive automated unit testing.
  - c. We achieve our sprint goals without working overtime.
9. Continuous attention to technical excellence and good design enhances agility.
  - a. We maintain automated unit test coverage on new code, and tests are all passing.
  - b. We use code coverage / static analysis tools to improve code quality.
  - c. There is a continuous build (or close to it) and automated regression tests.
10. Simplicity -- the art of maximizing the amount of work not done -- is essential.
  - a. We take the time to refactor in order to clean our code.
  - b. We build the simplest solution that will work first, then expand later as needed.

- c. Our User Stories are INVESTed and 3-day sized efforts or less.
- 11. The best architectures, requirements, and designs emerge from self-organizing teams.
  - a. Our team determines how we design and develop our software.
  - b. Our User Stories provide what is needed and why, and leave the how to the team.
  - c. We are self-organized, empowered to make decisions, and independently determine how to meet our sprint goals.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
  - a. We hold a retrospective after each sprint that results in a positive impact on the next sprint.
  - b. The challenges we incorporate are making us a more effective team.
  - c. During daily stand up, we identify opportunities for improvement.

## The 12 Principles behind the Agile Manifesto

We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity -- the art of maximizing the amount of work not done -- is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.