# Agile principles of teamwork

Teamwork is critical to agile projects. Creating good products requires cooperation among all the members of the project team, including customers and stakeholders. Agile approaches support team-building and teamwork, and they emphasize trust

Although all 12 principles support the goal of teamwork, principles 4–6, 8, 11, and 12 stand out for us as supporting team empowerment, efficiency, and excellence:

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.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
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.

Agile approaches focus on [illegible]

Here are some practices you can adopt to make this vision of teamwork a reality:

- Ensure that your development team members have the proper skills and motivation.
- Provide training sufficient to the task.
- Support the self-organizing development team's decisions about what to do and how to do it; don't have managers tell the team what to do.
- Hold project team members responsible as a single team, not individuals.
- Use face-to-face communication to quickly and efficiently convey information.

RNING

Suppose that you usually communicate by email to Sharon. You take time to craft your message and then send it. The message sits in Sharon's inbox, and she eventually reads it. If Sharon has any questions, she writes an email in response and sends it. That message sits in your inbox until you eventually read it. And so forth. This type of table tennis communication is too inefficient to use in the middle of a rapid iteration.

- Have spontaneous conversations throughout the day to build knowledge, understanding, and efficiency.
- Collocate teammates in close proximity to increase clear and efficient communication. If collocation isn't possible, use video chat rather than email.
- Make sure that *lessons learned* is an ongoing feedback loop. Retrospectives should be held at the end of each iteration, when reflection and adaptation can improve development team productivity going forward, creating ever higher levels of efficiency. A lessons learned meeting at the end of a project is of minimal value.

The first retrospective is often the most valuable because, at that point, the project team has the opportunity to make changes to benefit the rest of the project moving forward.

project moving forward.

The following strategies promote effective teamwork:

- Place the development team in the same location — this is called *collocation.*
- Put together a physical environment that's conducive for collaboration: a team room with whiteboards, colored pens, and other tactile tools for developing and conveying ideas to ensure shared understanding.
- Create an environment where project team members are encouraged to speak their minds.
- Meet face-to-face whenever possible. Don't send an email if a conversation can handle the issue.
- Get clarifications throughout the day as they're needed.
- Encourage the development team to solve problems rather than having managers solve problems for the development team.

# Agile principles of project management

Agility in project management encompasses three key areas:

- Making sure the development team can be productive and can sustainably increase productivity over long periods of time
- Ensuring that information about the project's progress is available to stakeholders without interrupting the flow of development activities by asking the development team for updates
- Handling requests for new features as they occur and integrating them into the product development cycle

All 12 principles support project management, but principles 2, 8, and 10 stand out for us:

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

10. Simplicity — the art of maximizing the amount of work not done — is essential.

Following are some advantages of adopting agile project management:

- Agile project teams achieve faster time-to-market, and consequentially cost savings. They start development earlier than in traditional approaches because agile approaches minimize the exhaustive upfront planning and documentation that is conventionally part of the early stages of a waterfall project.
- Agile development teams are self-organizing and self-managing. The managerial effort normally put into telling developers how to do their work can be applied to removing impediments and organizational distractions that slow down the development team.
- Agile development teams determine how much work they can accomplish in an iteration and commit to achieving those goals. Ownership is fundamentally different because the development team is establishing the commitment, not complying with an externally developed commitment.
- An agile approach asks, "What is the minimum we can do to achieve the goal?" instead of focusing on including all the features and extra refinements that could possibly be needed. An agile approach usually means streamlining: barely sufficient documentation, removal of unnecessary meetings, avoidance of inefficient communication (such as email), and less coding (just enough to make it work).

  Creating complicated documents that aren't useful for product development is a waste of effort. It's okay to document a decision, but you don't need multiple pages on the history and nuances of how the decision was made. Keep the documentation barely sufficient, and you will have more time to focus on supporting the development team.

- By encapsulating development into short sprints that last one to four weeks or less, you can adhere to the goals of the current iteration while accommodating change in subsequent iterations. The length of each sprint remains the same throughout the project to provide a predictable rhythm of development for the team long-term.

- Planning, elaborating on requirements, developing, testing, and demonstrating functionality occur within an iteration, lowering the risk of heading in the wrong direction for extended periods of time or developing something that the customer doesn't want.

- Agile practices encourage a steady pace of development that is productive and healthy. For example, in the popular agile development set of practices called extreme programming (XP), the maximum workweek is 40 hours, and the preferred workweek is 35 hours. Agile projects are sustainable and more productive, especially long term.

  Traditional approaches routinely feature a *death march,* in which the project team puts in extremely long hours for days and even weeks at the end of a project to meet a previously unidentified and unrealistic deadline. As the death march goes on, productivity tends to drop dramatically. More defects are introduced, and because defects need to be corrected in a way that doesn't break a different piece of functionality, correcting defects is the most expensive work that can be performed. Defects are often the result of overloading a system — specifically demanding an unsustainable pace of work. Check out our presentation on the negative effects of "Racing in Reverse" (`https://platinumedge.com/overtime`).

- Priorities, experience on the existing project, and, eventually, the speed at which development will likely occur within each sprint are clear, making for good decisions about how much can or should be accomplished in a given amount of time.

**TABLE 2-4** Contrasting Historical Project Management with Agile Project Management

| Traditional Project Management Tasks | Agile Approach to the Project Management Task |
|---|---|
| Create a fully detailed project requirement document at the beginning of the project. Try to control requirement changes throughout the project. | Create a product backlog — a simple list of requirements by priority. Quickly update the product backlog as requirements and priorities change throughout the project. |
| Conduct weekly status meetings with all project stakeholders and developers. Send out detailed meeting notes and status reports after each meeting. | The development team meets quickly, for no longer than 15 minutes, at the start of each day to coordinate and synchronize that day's work and any roadblocks. They can update the centrally visible burndown chart in under a minute at the end of each day. |
| Create a detailed project schedule with all tasks at the beginning of the project. Try to keep the project tasks on schedule. Update the schedule on a regular basis. | Work within sprints and identify only specific tasks for the active sprint. |
| Assign tasks to the development team. | Support the development team by helping remove impediments and distractions. On agile projects, development teams define and pull (as opposed to push) their own tasks. |