

FIGURE 6-1:
Agile project team, scrum team, and development team.

Drew

Product owner

The product owner, sometimes called the *customer representative* in non-scrum environments, is responsible for bridging the gaps between the customer, business stakeholders, and the development team. The product owner is an expert on the product and the customer's needs and priorities. The product owner, who is a peer member of the scrum team, shields the development team from business distractions, works with the development team daily to help clarify requirements, and accepts completed work throughout the sprint in preparation for the sprint review.

On an agile project, the product owner will

- » Develop strategy and direction for the project and set long- and short-term goals.
- » Provide or have access to product expertise.
- » Understand and convey the customer's and other business stakeholders' needs to the development team.
- » Gather, prioritize, and manage product requirements.
- » Take responsibility for the product's budget and profitability.
- » Decide when to release completed functionality.
- » Work with the development team on a daily **basis to answer** questions and make decisions.
- » Accept or reject completed work — as it's **completed** — during the sprint.
- » Present the scrum team's accomplishments **at the end** of each sprint, before the development team demonstrates **these accomplishments**.

Development team member

Development team members are the people who create the product. In software development, programmers, testers, designers, writers, data engineers, and anyone else with a hands-on role in product development are development team members. With other types of product, the development team members may have different skills.

On an agile project, the development team is

- » Directly accountable for creating project deliverables.
- » Self-organizing and self-managing. The development team members determine their own tasks and how they want to complete those tasks.
- » Cross-functional. Collectively, the development team possesses all skills required to elaborate, design, develop, test, integrate and document requirements into working functionality.
- » Multi-skilled. Development team members are versatile — they're not tied to a single skill set. They have existing skills to immediately contribute at the beginning of the project, but they are also willing to learn new skills and to teach what they know to other development team members.
- » Ideally dedicated to one project for the duration of the project.
- » Ideally collocated. The team should be working together in the same area of the same office.

What makes a good development team member? Take a look at the team respon-

Scrum master

A scrum master, sometimes called a *project facilitator* in non-scrum agile environments, is responsible for supporting the development team, clearing organizational roadblocks, and keeping processes true to agile principles.

On an agile project, the scrum master will

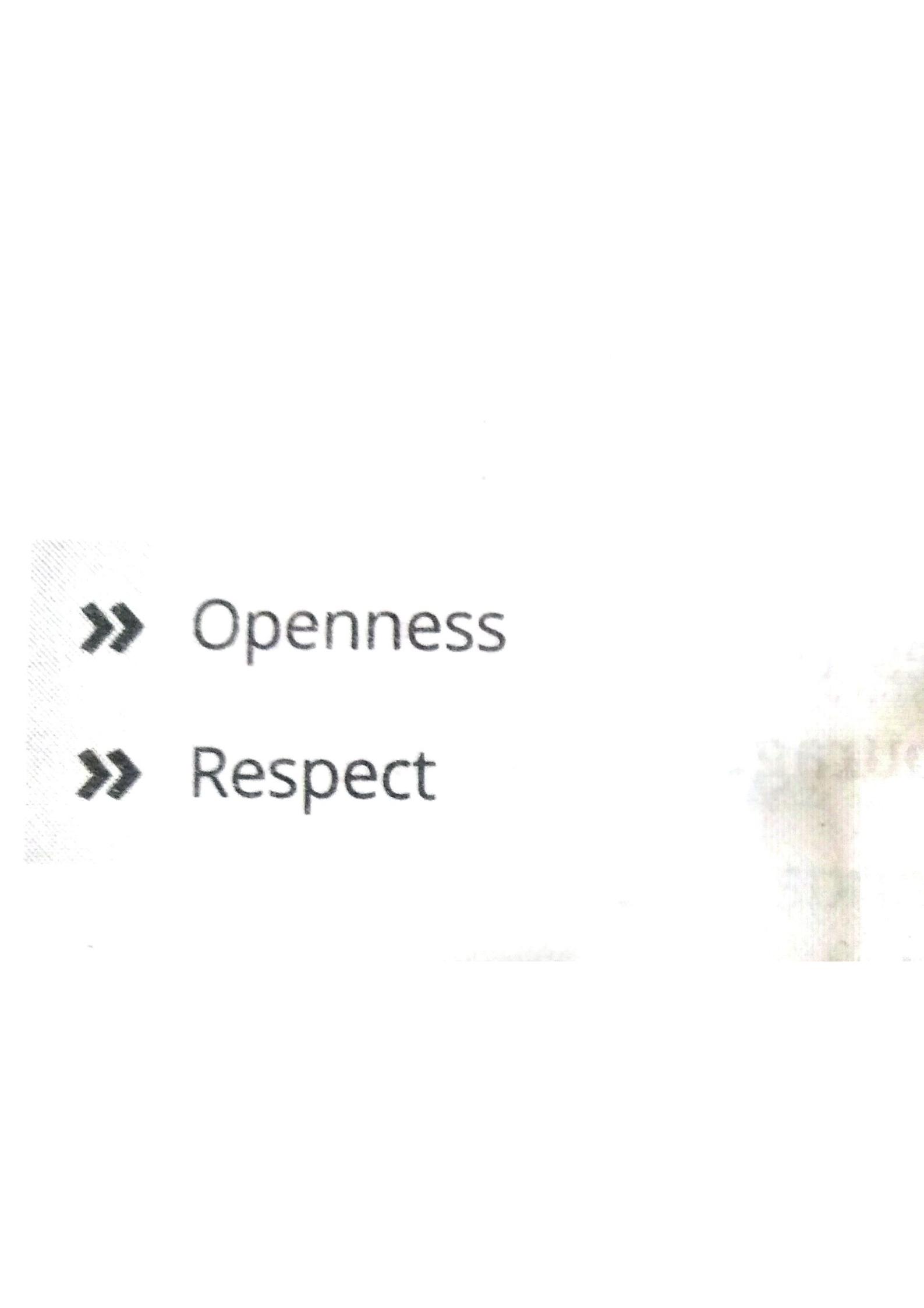
- » Act as a process coach, helping the project team and the organization follow scrum values and practices.
- » Help remove project impediments — both reactively and proactively — and shield the development team from external interferences.
- » Foster close cooperation between stakeholders and the scrum team.
- » Facilitate consensus building within the scrum team.
- » Protect the scrum team from organizational distractions.

Establishing New Values

Lots of organizations post their core values on the wall. In this section, however, we are talking about values that represent a way of working together every day, supporting each other, and doing whatever it takes to achieve the scrum team's commitments.

In addition to the values from the Agile Manifesto, the five core values for scrum teams are

- » Commitment
- » Courage
- » Focus



» Openness

» Respect

Commitment



- » Scrum teams must be realistic when making commitments, especially for short sprints. It is easier, both logistically and psychologically, to bring new features into a sprint than it is to take unachievable features out of a sprint.
- » Scrum teams must fully commit to goals. This includes having consensus among the team that the goal is achievable. After the scrum team agrees on a goal, the team does whatever it takes to reach that goal.
- » The scrum team is pragmatic but ensures that every sprint has a tangible value. Achieving a sprint goal and completing every item in the goal's scope are different. For example, a sprint goal of proving that a product can perform a specific action is much better than a goal stating that exactly seven requirements will be complete during the sprint. Effective scrum teams focus on the goal and remain flexible in the specifics of how to reach that goal.
- » Scrum teams are willing to be accountable for results. The scrum team has the power to be in charge of the project. As a scrum team member, you can be responsible for how you organize your day, the day-to-day work, and the outcome.

Courage

- » **Realize that the processes that worked in the past won't necessarily work now.** Sometimes you need to remind people of this fact. If you want to be successful with agile techniques, your everyday work processes need to change to improve.
- » **Be ready to buck the status quo.** The status quo will push back. Some people have vested interests and will not want to change how they work.
- » **Temper challenge with respect.** Senior members of the organization might be especially resistant to change; they often created the old rules for how things were done. Now you're challenging those rules. Respectfully remind these individuals that you can achieve the benefits of agile techniques only by following the 12 agile principles faithfully. Ask them to give change a try.
- » **Embrace the other values.** Have the courage to make commitments and stand behind those commitments. Have the courage to focus and tell distractors "no." Have the courage to be open and to acknowledge that there is always an opportunity to improve. And have the courage to be respectful and tolerant of other people's views, even when they challenge your views.



Focus

- » **Ensure that you're not spending time on activities unrelated to the sprint goal.** If someone tries to distract you from the sprint goal with something that "has to be done," explain your priorities. Ask, "How will this request move the sprint goal forward?" This simple question can push a lot of activities off the to-do list.
- » **Figure out what needs to be done and do only that.** The development team determines the tasks necessary to achieve the sprint goal. If you're a development team member, use this ownership to drive your focus to the priority tasks at hand.
- » **Balance focused time with accessibility to the rest of the scrum team.** Francesco Cirillo's Pomodoro technique — splitting work into 25-minute time blocks, with breaks in between — helps achieve balance between focus and accessibility. We often recommend giving development team members noise-canceling headsets, the wearing of which is a "do not disturb" sign. However, we also suggest a team agreement that all scrum team members have a minimum set of office hours in which they are available for collaboration.
- » **Check that you're maintaining your focus.** If you're unsure of whether you are maintaining focus — it can be hard to tell — go back to the basic question, "Are my actions consistent with achieving the overall goal and the near-term goal (such as completing the current task)?"



Openness

- » **Be open and encourage openness in others.** Team members must feel free to speak openly about problems and opportunities to improve, whether the issues are something that they're dealing with themselves or see elsewhere in the team. Openness requires trust within the team, and trust takes time to develop.
- » **Defuse internal politics by discouraging gossip.** If someone starts talking to you about what another team member did or didn't do, ask him or her to take the issue to the person who can resolve it. Don't gossip yourself. Ever.
- » **Always be respectful.** Openness is never an excuse to be destructive or mean. Respect is critical to an open team environment.

Respect

Each individual on the team has something important to contribute. Your background, education, and experiences have a distinctive influence on the team. Share your uniqueness and look for, and appreciate, the same in others. You encourage respect when you

- » **Foster openness.** Respect and openness go hand in hand. Openness without respect causes resentment; openness with respect generates trust.
- » **Encourage a positive work environment.** Happy people tend to treat one another better. Encourage positivity, and respect will follow.
- » **Seek out differences.** Don't just tolerate differences; try to find them. The best solutions come from diverse opinions that have been considered and appropriately challenged.
- » **Treat everyone on the team with the same degree of respect.** All team members should be accorded the same respect, regardless of their role, level of experience, or immediate contribution. Encourage everyone to give his or her best.

Changing Team Philosophy

- » **Dedicated team:** Each scrum team member works only on the project assigned to the scrum team, and not with outside teams or projects. Projects may finish and new projects may start, but the team stays the same.
- » **Cross-functionality:** The willingness and ability to work on different types of tasks to create the product.
- » **Self-organization:** The ability and responsibility to determine how to go about the work of product development.
- » **Self-management:** The ability and responsibility to keep work on track.
- » **Size-limited teams:** Right-size development teams to ensure effective communication. Smaller is better; the development team should never be larger than nine people.
- » **Ownership:** Take initiative for work and responsibility for results.

Dedicated team

A traditional approach to resource allocation (we prefer the term *talent allocation*) is to allocate portions of team members' time across multiple teams and projects to get to full 100 percent utilization to justify the expense of employing team members. For management, knowing that all hours of the week are accounted for and justified is gratifying. However, the result is lower productivity due to continual *context switching* — the cost associated with cognitive demobilization and remobilization to switch from one task to another.

The following results occur when you dedicate scrum teams to work on only one project at a time:

- » **More accurate release projections:** Because the same people are consistently doing the same tasks every sprint with the same amount of time allocated to the project from sprint to sprint, scrum teams can accurately and empirically extrapolate how long it will take to complete their remaining backlog items with more certainty than traditional splintered approaches.
- » **Effective, short iterations:** Sprints are short because the shorter the feedback loop, the more quickly scrum teams can respond to feedback and changing needs. There just isn't enough time for thrashing team members between projects.
- » **Fewer and less costly defects:** Context switching results in more defects because distracted developers produce lower quality functionality. It costs less to fix something while it is still fresh in your mind (during the sprint) than later, when you have to try to remember the context of what you were working on. Studies show that defects cost 6.5 times more to fix after the sprint ends and you've moved on to other requirements, 24 times more to fix when preparing for release, and 100 times more to fix after the product is in production.

Cross-functionality

Cross-functionality encourages each team member to

- » **Set aside the narrow label of what he or she can do.** Titles have no place on an agile team. Skills and an ability to contribute are what matter. Start thinking of yourself as a Special Forces commando — knowledgeable enough in different areas that you can take on any situation.
- » **Work to expand skills.** Don't work only in areas you already know. Try to learn something new each sprint. Techniques such as *pair programming* — where two developers work together to code one item — or shadowing other developers can help you learn new skills quickly and increase overall product quality.
- » **Step up to help someone who has run into a roadblock.** Helping someone with a real-world problem is a great way to learn a new skill.
- » **Be flexible.** A willingness to be flexible helps to balance workloads and makes the team more likely to reach its sprint goal.

Self-organization

Agile techniques emphasize self-organizing development teams to take advantage of development team members' varied knowledge and experience.

- » **Commit to their own sprint goals.** At the beginning of each sprint, the development team works with the product owner to identify an objective it can reach, based on project priorities.
- » **Identify their tasks.** Development team members determine the tasks necessary to meet each sprint goal. The development team works together to figure out who takes on which task, how to get the work done, and how to address risks and issues.
- » **Estimate the effort necessary for requirements and related tasks.** The development team knows the most about how much effort it will take to create specific product features.
- » **Focus on communication.** Successful agile development teams hone their communication skills by being transparent, communicating face-to-face, being aware of nonverbal communication, participating, and listening.

The key to communication is clarity. With complex topics, avoid one-way, potentially ambiguous modes of communication, such as email. Face-to-face communication prevents misunderstandings and frustration. You can always summarize the conversation in a quick email later if details need to be retained.
- » **Collaborate.** Getting the input of a diverse scrum team almost always improves the product but requires solid collaboration skills. Collaboration is the foundation of an effective agile team.

No successful project is an island. Collaboration skills help scrum team members take risks with ideas and bring innovative solutions to project problems. A safe and comfortable environment is a cornerstone of a successful agile project.

- » **Decide with consensus.** For maximum productivity, the entire development team must be on the same page and committed to the goal at hand. The scrum master often plays an active role in building consensus, but the development team ultimately takes responsibility for reaching agreement on decisions, and everyone owns the decisions.
- » **Actively participate.** Self-organization may be challenging for the shy. All development team members must actively participate. No one is going to tell the development team what to do to create the product. The development team members tell themselves what to do. And when. And how.

City Indore

Self-management

- » **Allow leadership to ebb and flow.** On agile projects, each person on the development team has the opportunity to lead. For different tasks, different leaders will naturally emerge; leadership will shift throughout the team based on skill expertise and previous experiences.
- » **Rely on agile processes and tools to manage the work.** Agile methods are tailored to make self-management easy. With an agile approach, meetings have clear purposes and time limits, and artifacts expose information but rely on minimal effort to create and maintain. Taking advantage of these processes allows development teams to spend most of their time creating the product.

- » **Report progress regularly and transparently.** Each development team member is responsible for accurately updating work status on a daily basis. Luckily, progress reporting is a quick task on agile projects. In Chapter 9, you find out about burndown charts, which provide status but only require a few minutes each day to update. Keeping status current and truthful makes planning and issue management easier.
- » **Manage issues within the development team.** Many obstacles can arise on a project: Development challenges and interpersonal problems are a couple of examples. The development team's first point of escalation for most issues is the development team itself.
- » **Create a team agreement.** Development teams sometimes make up a team agreement, a document that outlines the expectations each team member will commit to meet. Working agreements provide a shared understanding of behavioral expectations and empower the facilitator to keep the team on track according to what they've already agreed together.
- » **Inspect and adapt.** Figure out what works for your team. Best practices differ from team to team. Some teams work best by coming in early; others work best by coming in late. The development team is responsible for reviewing its own performance and identifying techniques to continue and techniques to change.
- » **Actively participate.** As with self-organization, self-management works only when development team members join in and commit to guiding the project's

Size-limited teams

Limiting development team size

- » Encourages diverse skills to be developed
- » Facilitates good team communication
- » Maintains the team in a single unit
- » Promotes joint code ownership, cross-functionality, and face-to-face communication

Ownership

Being part of a cross-functional, self-organized, self-managing development team requires responsibility and ownership. The top-down management

Development teams can adapt behavior and increase their level of ownership by doing the following:

- » **Take initiative.** Instead of waiting for someone else to tell you what to work on, take action. Do what is necessary to help meet commitments and goals.
- » **Succeed and fail as a team.** On agile projects, accomplishments and failures alike belong to the project team. If problems arise, be accountable as a group,

rather than finding blame. When you succeed, recognize the group effort necessary for that success.

- » **Trust the ability to make good decisions.** Development teams can make mature, responsible, and sound decisions about product development. This takes a degree of trust as team members become accustomed to having more control in a project.

Behavioral maturity