

Being Agile

Agile Approaches – Environments and
Behaviours, Techniques



**SILESIAN
UNIVERSITY**

SCHOOL OF BUSINESS
ADMINISTRATION IN KARVINA

Pavel Adámek
Project management



- Agile Approaches
 - Reviewing the Big Three: Lean, Scrum, and Extreme Programming
 - Agile Environments in Action
 - Agile Behaviours in Action
-



Agile is a descriptive term for a number of techniques and methods that have the following similarities:

- Development within multiple iterations, called iterative development
 - Emphasis on simplicity, transparency, and situation-specific strategies
 - Cross-functional, self-organizing teams
 - Working functionality as the measure of progress
-

With regards to product development, the empirical approach is braced by the following pillars:

- **Unfettered transparency:** Everyone involved in the process understands and can contribute to the development of the process.
 - **Frequent inspection:** The inspector must inspect the product regularly and possess the skills to identify variances from acceptance criteria.
 - **Immediate adaptation:** The development team must be able to adjust quickly to minimize further product deviations.
-



Three, however, are common to many agile projects: lean product development, scrum, and extreme programming (XP).

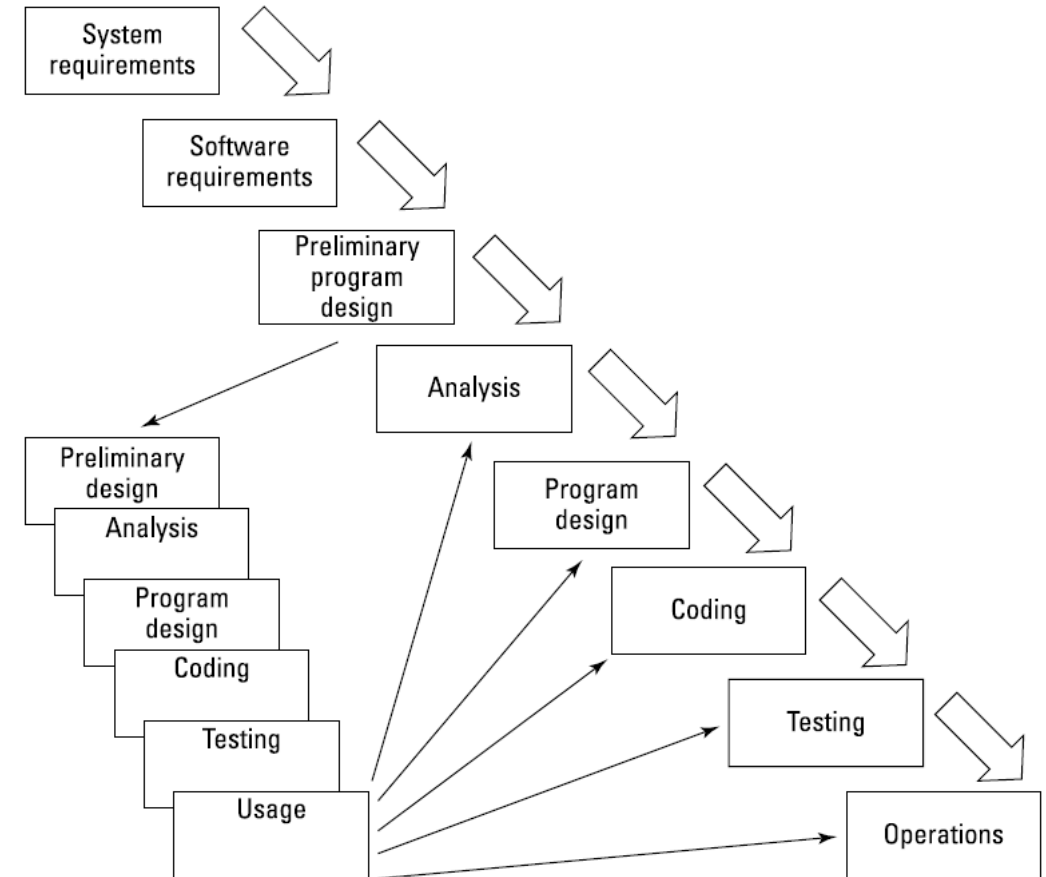
These three approaches work perfectly together and share many common elements, although they use different terminology or have a slightly different focus.

Broadly, lean and scrum focus on structure.

Extreme programming does that, too, but is more prescriptive about development practices, focusing more on technical design, coding, testing, and integration.

- Dr. Winston Royce in his article, “*Managing the Development of Large Systems*,” published in 1970, Dr. Royce codified the step-by-step software development process known as waterfall.
- Hardware became repeatable through mass production, and software became the more complex and diverse aspect of a complete solution.
- The irony here is that, even though the diagram implies that you complete tasks step by step, Dr. Royce himself added the cautionary note that you need **iteration**.

Iteration in waterfall



Agile Approaches - Reviewing the Big Three: Lean, Scrum, and Extreme Programming

An overview of lean, Scrum and XP

- Lean has its origins in manufacturing. Mass production methods, which have been around for more than 100 years, were designed to simplify assembly processes (for example, putting together a Model-T Ford).
- These processes use complex, expensive machinery and low-skilled workers to inexpensively churn out an item of value. The idea is that if you keep the machines and people working and stockpile inventory, you generate a lot of efficiency.



An overview of lean



- Traditionally, mass production requires wasteful supporting systems and large amounts of indirect labour to ensure that manufacturing continues without pause. It generates a huge inventory of parts, extra workers, extra space, and complex processes that don't add direct value to the product. Sound familiar?
 - In the 1940s in Japan, a small company called Toyota wanted to produce cars for the Japanese market but couldn't afford the huge investment that mass production requires.
 - The company studied supermarkets, noting how consumers buy just what they need because they know there will always be a supply and how the stores restock shelves only as they empty.
 - From this observation, Toyota created a just-in-time process that it could translate to the factory floor.
 - The result was a significant reduction in inventory of parts and finished goods and a lower investment in machines, people, and space.
-



- The term lean was coined in the 1990s in **The Machine That Changed the World: The Story of Lean Production** (Free Press) by James P. Womack, Daniel T. Jones, and Daniel Roos.
 - **eBay** was an early adopter of lean principles for software development.
 - The company led the way with an approach that responded daily to customers' requests for changes to the website, developing high-value features in a short time.
 - The focus of lean is business value and minimizing activities outside product development.
-

Mary and Tom Poppendieck discuss a group of lean principles on their blog and in their books on lean software development. Following are the lean principles from their 2003 book, *Lean Software Development*:

- **Eliminate waste.** Doing anything that is beyond barely sufficient (process steps, artifacts, meetings) slows down the flow of progress. Waste includes failing to learn from work, building the wrong thing, and thrashing (context switching between tasks or projects) — which results in only partially creating lots of product features but not completely creating any.
 - **Amplify learning.** Learning drives predictability. Enable improvement through a mindset of regular and disciplined transparency, inspection, and adaptation.
 - **Deliver as late as possible.** Allow for late adaptation. Don't deliver late, but leave your options open long enough to make decisions at the last responsible moment based on facts rather than uncertainty — when you know the most. Learn from failure. Challenge standards. Use the scientific method — experiment with hypotheses to find solutions.
-

An overview of lean



- **Deliver as fast as possible.** Speed, cost, and quality are compatible. The sooner you deliver, the sooner you receive feedback. Work on fewer things at once, limiting work in progress and optimizing flow. Manage workflow, rather than schedules. Use just-in-time planning to shorten development and release cycles.
 - **Empower the team.** Working autonomously, mastering skills, and believing in the purpose of work can motivate development teams. Managers do not tell developers how to do their jobs but instead support them to self-organize around the work to be done and remove their impediments.
 - **Build quality in.** Establish mechanisms to catch and correct defects when they happen and before final verification. Quality is built in from the beginning, not at the end. Break dependencies, so you can develop and integrate functionality at any time without regressions.
 - **See the whole.** An entire system is only as strong as its weakest link. Solve problems, not just symptoms. Continually pay attention to bottlenecks throughout the flow of work and remove them.
-

An overview of lean



- Beyond the lean principles, one of the most common lean approaches used by agile teams is kanban, sometimes referred to as lean-kanban.

Kanban practices include the following:

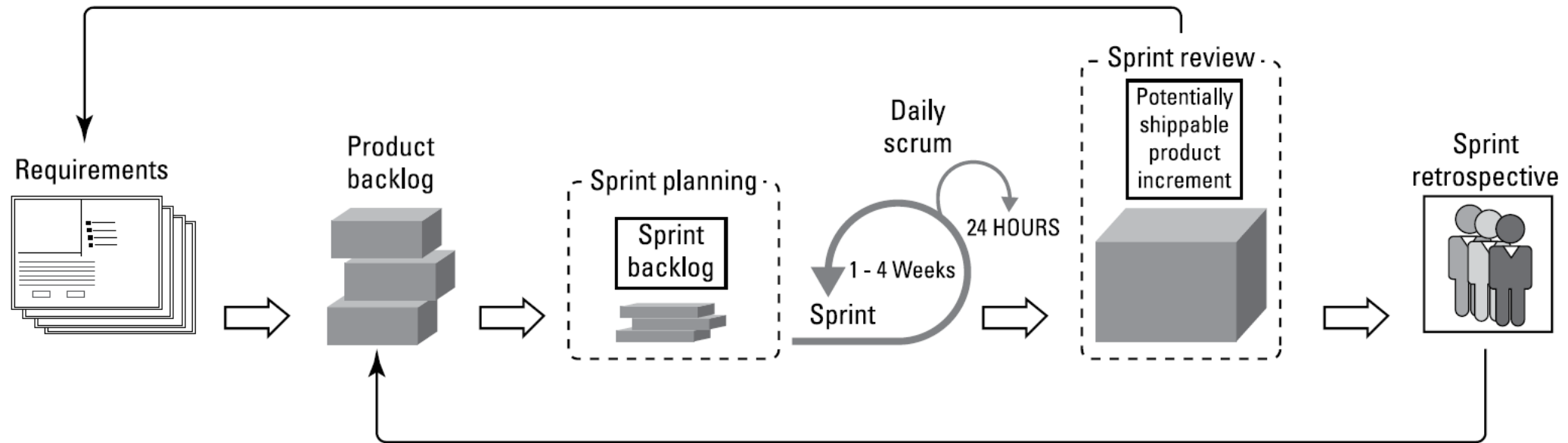
- Visualize - As agile teams visualize the flow of their work (on a whiteboard, on a wall, or in a drawing) and identify where productivity breaks down, they can easily analyze the root cause and see how to remove the constraint. And then do it again, and again, and again.
 - Limit work in progress (WIP) - Being agile is all about getting to done, so the goal is to start things only when other things are completed.
 - Manage flow - Measuring lead and cycle times helps agile teams monitor their management of flow. A team determines the lead time by tracking the amount of time it takes a request for functionality to go from arrival in the queue to complete.
 - Make process policies explicit.
 - Implement feedback loops.
 - Improve collaboratively, evolve experimentally.
-

An overview of scrum



- Scrum, the most popular agile framework in software development, is an iterative approach that has at its core the sprint (the scrum term for iteration). To support this process, scrum teams use specific roles, artifacts, and events. To make sure that they meet the goals of each part of the process, scrum teams use inspection and adaptation throughout the project.

Scrum approach



An overview of scrum



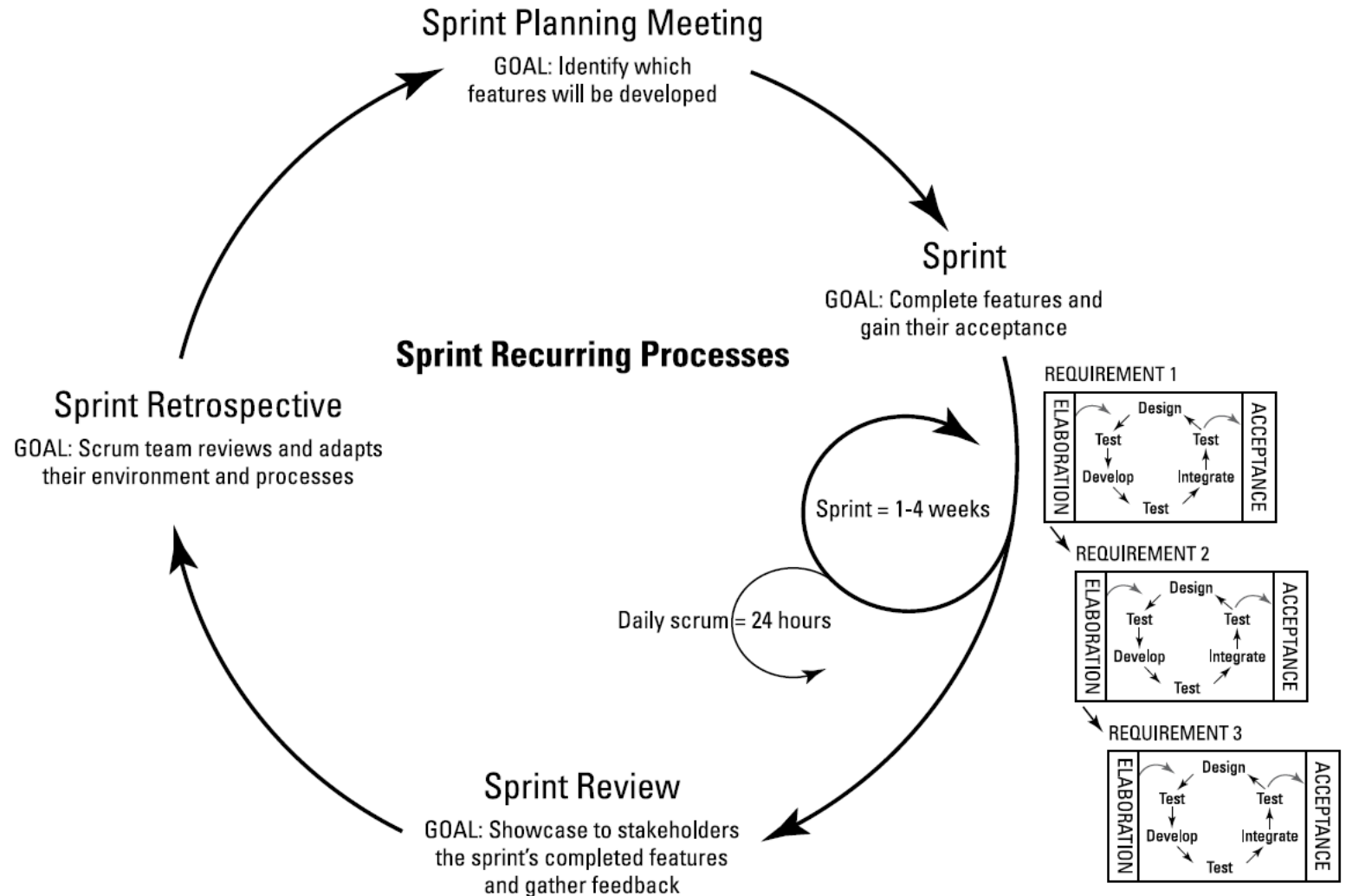
- Within each sprint, the development team develops and tests a functional part of the product until the product owner accepts it, often daily, and the functionality becomes a potentially shippable increment of the overall product.
 - When one sprint finishes, another sprint starts. Releasing functionality to the market often occurs at the end of multiple sprints, when the product owner determines that enough value exists.
 - A core principle of the sprint is its cyclical nature: The sprint, as well as the processes within it, repeats over and over.
 - You use the tenets of inspection and adaptation on a daily basis as part of a scrum project:
 - During a sprint, you conduct constant inspections to assess progress toward the sprint goal, and consequentially, toward the release goal.
 - You hold a daily scrum meeting to organize the day by reviewing what the team completed yesterday and coordinating what it will work on today. Essentially, the scrum team inspects its progress toward the sprint goal.
 - At the end of the sprint, you use a sprint retrospective meeting to assess performance and plan necessary adaptations.
-

An overview of scrum



- These inspections and adaptations may sound formal and process-laden, but they aren't.
- Use inspection and adaptation to solve issues and don't overthink this process.
- The problem you're trying to solve today will often change in the future anyway.

Sprints are recurring processes





Understanding scrum roles, artifacts, and events.

- The scrum framework defines specific roles, artifacts, and events for projects.

Scrum's three roles — people on the project — are as follows:

- **Product owner:** Represents and speaks for the business needs of the project.
 - **Development team:** Performs the day-to-day work. The development team is dedicated to the project and each team member is multi-skilled — that is, although team members may have certain strengths, each member is capable of doing multiple jobs on the project.
 - **Scrum master:** Protects the team from organizational distractions, clears roadblocks, ensures that scrum is played properly, and continuously improves the team's environment.
-

An overview of scrum



Additionally, scrum teams find that they're more effective and efficient when they work closely with two non-scrum-specific roles:

- **Stakeholders:** Anyone who is affected by or has input on the project. Although stakeholders are not official scrum roles, it is essential for scrum teams and stakeholders to work closely together throughout a project.
 - **Agile mentor:** An experienced authority on agile techniques and the scrum framework. Often this person is external to the project's department or organization, so he or she can support the scrum team objectively with an outsider's point of view.
-

In the same way that scrum has specific roles, scrum also has three tangible deliverables, called **artifacts**:

- **Product backlog:** The full list of requirements that defines the product, often documented in terms of business value from the perspective of the end user. The product backlog can be fluid throughout the project. All scope items, regardless of level of detail, are in the product backlog. The product owner owns the product backlog, determining what goes in it and in what priority.
 - **Sprint backlog:** The list of requirements and tasks in a given sprint. The product owner and the development team select the requirements for the sprint in sprint planning, with the development team breaking down these requirements into tasks. Unlike the product backlog, the sprint backlog can be changed only by the development team.
 - **Product increment:** The usable, potentially shippable functionality. Whether the product is a website or a new house, the product increment should be complete enough to demonstrate its working functionality. A scrum project is complete after a product contains enough shippable functionality to meet the customer's business goals for the project.
-

Scrum also has five events:

- **Sprint:** Scrum's term for iteration. The sprint is the container for each of the other scrum events, in which the scrum team creates potentially shippable functionality. Sprints are short cycles, no longer than a month, typically between one and two weeks, and in some cases as short as one day. Consistent sprint length reduces variance; a scrum team can confidently extrapolate what it can do in each sprint based on what it has accomplished in previous sprints. Sprints give scrum teams the opportunity to make adjustments for continuous improvement immediately, rather than at the end of the project.
 - **Sprint planning:** Takes place at the start of each sprint. In sprint planning meetings, scrum teams decide which goal, scope, and supporting tasks will be part of the sprint backlog.
 - **Daily scrum:** Takes place daily for no more than 15 minutes. During the daily scrum, development team members make three statements:
 - What the team member completed yesterday
 - What the team member will work on today
 - A list of items impeding the team member
-

An overview of scrum



Scrum also has events:

- **Sprint review:** Takes place at the end of each sprint. In this meeting, the development team demonstrates to the stakeholders and the entire organization the accepted parts of the product the team completed during the sprint. The key to the sprint review is collecting feedback from the stakeholders, which informs the product owner how to update the product backlog and consider the next sprint goal.
 - **Sprint retrospective:** Takes place at the end of each sprint. The sprint retrospective is an internal team meeting in which the scrum team members (product owner, development team, and scrum master) discuss what went well during the sprint, what didn't work well, and how they can make improvements for the next sprint.
 - This meeting is action-oriented (frustrations should be vented elsewhere) and ends with tangible improvement plans for the next sprint.
-

An overview of extreme programming



- One popular approach to product development, specific to software, is extreme programming (XP). Extreme programming takes the best practices of software development to an extreme level. Created in 1996 by Kent Beck, with the help of Ward Cunningham and Ron Jeffries, the principles of XP were originally described in Beck's 1999 book, *Extreme Programming Explained*.
 - The focus of extreme programming is customer satisfaction. XP teams achieve high customer satisfaction by developing features when the customer needs them.
 - New requests are part of the development team's daily routine, and the team is empowered to deal with these requests whenever they crop up. The team organizes itself around any problem that arises and solves it as efficiently as possible.
-

Agile Environments in Action

Agile project teams flourish when scrum team members work closely together in an environment that supports the process.

- The development team members are central to the success of agile projects.
- Creating the right environment for them to operate in goes a long way toward supporting their success.
- You can even hire people who specialize in designing optimal agile work environments.

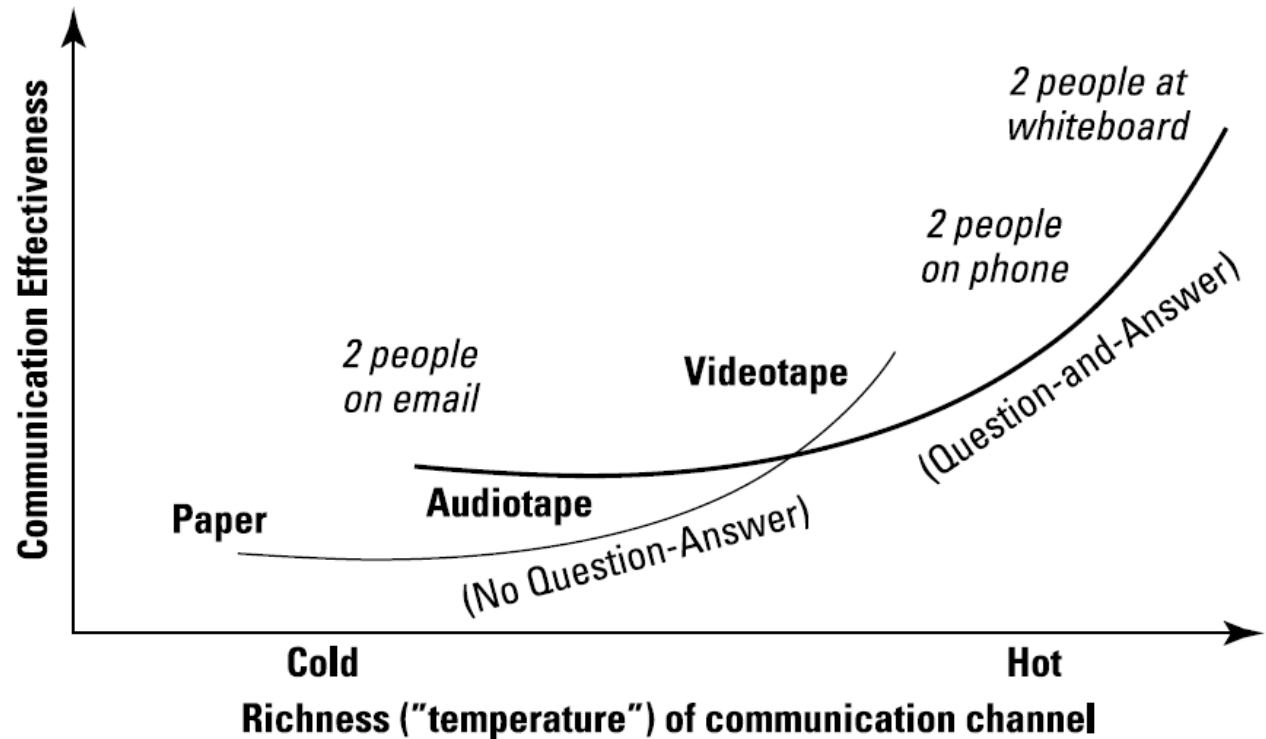


When a scrum team is collocated, the following practices are possible and significantly increase efficiency and effectiveness:

- Communicating face to face
 - Physically standing up — rather than sitting — as a group for the daily scrum meeting (this keeps meetings brief and on topic)
 - Using simple, low-tech tools for communication
 - Getting real-time clarifications from scrum team members
 - Being aware of what others are working on
 - Asking for help with a task
 - Supporting others with their tasks
-

- Alistair Cockburn, one of the Agile Manifesto signatories, created the graph shows the effectiveness of different forms of communication.
- Notice the difference in effectiveness between paper communication and two people at a whiteboard - with collocation, you get the benefit of better communication.

Better communication through collocation





- Set up an environment where the scrum team can work in close physical proximity.
 - If possible, the scrum team should have its own room, sometimes called a project room or a scrum room. The scrum team members create the setup they need in this project room, putting whiteboards and bulletin boards on the walls and moving the furniture.
 - Agile project teams take a responsive approach, and scrum team members require an environment that helps them respond to the project needs of the day. An agile team Environment should be mobile — literally:
 - Use movable desks and chairs so that people can move about and reconfigure the space.
 - Get wirelessly connected laptops so that scrum team members can pick them up and move them about easily.
 - Have a large mobile whiteboard. Also see the next section on low-tech communication.
-

- Rely on face-to-face conversations and good old-fashioned pen and paper. Low-tech promotes informality, allowing scrum team members to feel that they can change work processes and be innovative as they learn about the product.
- The primary tool for communication should be **face-to-face conversation**.

Tackling problems in person is the best way to accelerate production:

- Have short daily scrum meetings in person. Some scrum teams stand throughout a meeting to discourage it from running longer than 15 minutes.
 - Ask the product owner questions. Also, make sure he or she is involved in discussions about product features to provide clarity when necessary. The conversation shouldn't end when planning ends.
 - Communicate with your co-workers. If you have questions about features, the project's progress, or integrating, communicate with co-workers. The entire development team is responsible for creating the product, and team members need to talk throughout the day.
-

Agile Environments in Action

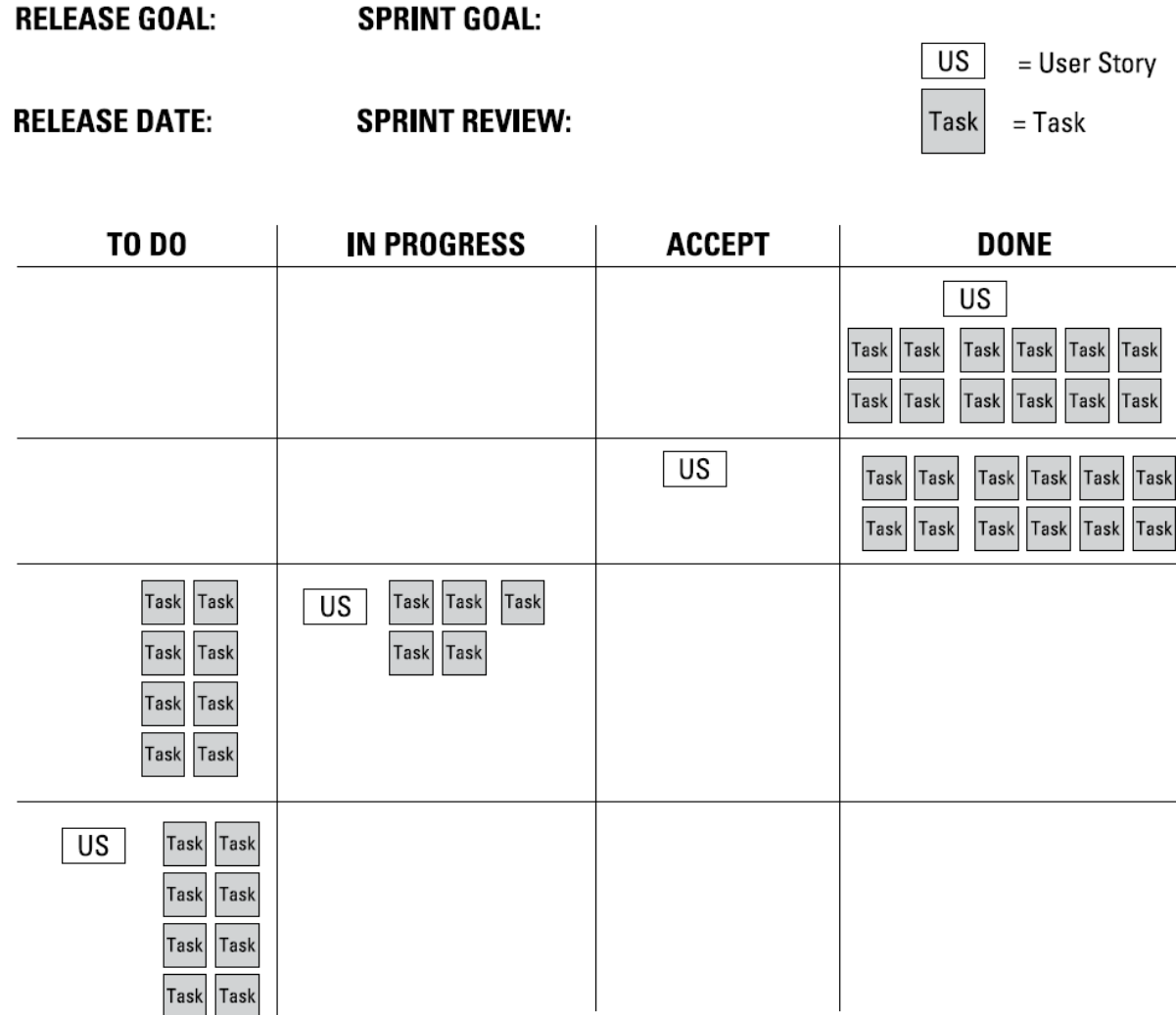


FIGURE 5-2:
A scrum task board on a wall or whiteboard.

- Basically, you move sticky notes or cards around the board to show the status.
- Everyone knows how to read the board and how to act on what it shows.
- Whatever tools you use, avoid spending time making things look perfectly neat and pretty.
- Formality in layout and presentation (what you might call pageantry) can give an impression that the work is tidy and elegant. However, the work is what matters, so focus your energy on activities that support the work.

Agile Behaviours in Action

Establishing Agile Roles

- The behavioural dynamics that need to shift for your organization to benefit from the performance advantages that agile techniques enable.
- You find out about the different roles on an agile project and see how you can change a project team's values and philosophy about project management.



The scrum framework defines common agile roles in an especially succinct manner. We use scrum terms to describe agile roles throughout this book. These roles are:

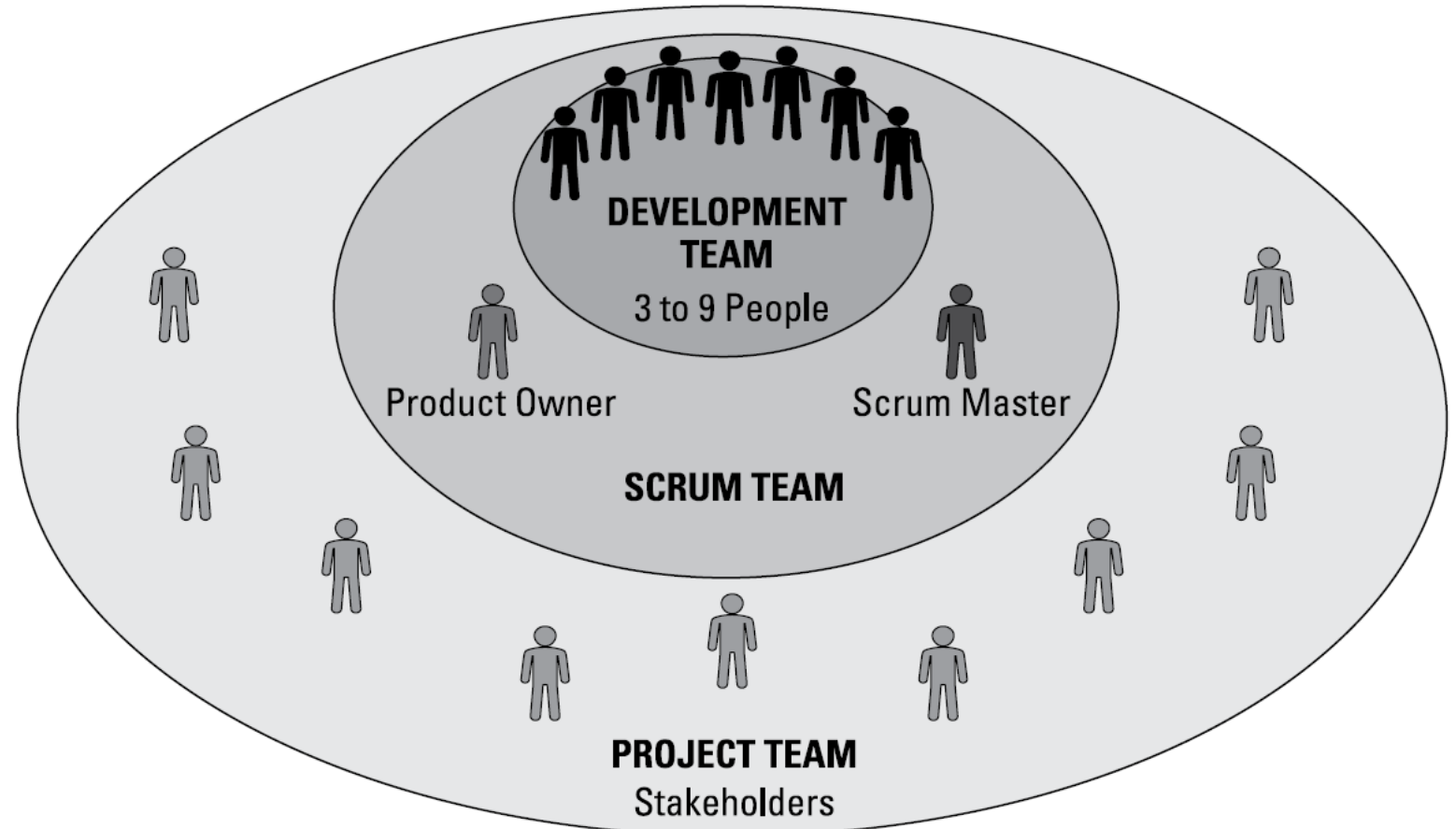
- Product owner
- Development team member
- Scrum master

- The product owner, development team, and scrum master together make up the scrum team. Each role is a peer to the others — no one is the boss of anyone else on the team.

The following roles are not part of the scrum framework but are still critically important to agile projects:

- Stakeholders
 - Agile mentor
-

- The scrum team together with the stakeholders make up the agile project team.
- At the center of it all is the development team.
- The product owner and scrum master fulfill roles that ensure the development team's success.





- 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.

On an agile project, the product owner will

- Develop strategy and direction for the project and set long- and shortterm 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.
-

Agile Behaviours in Action



- A scrum master is different from a project manager. Teams using traditional project approaches work for a project manager.
- A scrum master, on the other hand, is a servant-leader peer who supports the team so that it is fully functional and productive.
- The scrum master role is an enabling role, rather than an accountability role.

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.
-

To support good product development practices, remember the following:

- **Don't develop features that you're unlikely to use.**
 - **Make the development team central to the project because it adds the biggest value.**
 - **Have the customers prioritize features — they know what's most important to them. Tackle high-priority items first to deliver value.**
 - **Use tools that support great communication across all parties.**

 - **Scrum is simple: three roles, three artifacts, and five events. Each plays a part to ensure that the scrum team has continuous transparency, inspection, and adaptation throughout the project. As a framework, scrum accommodates many other agile techniques, methods, and tools for executing the technical aspects of building functionality.**
-

RECAP



- **An agile development team operates differently from a team using a waterfall approach. Development team members must change their roles based on each day's priorities, organize themselves, and think about projects in a whole new way to achieve their commitments.**

To be part of a successful agile project, development teams should embrace the following attributes:

- **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.
-