Scrum Team

Dr. Elham Mahmoudzadeh

The Scrum Team

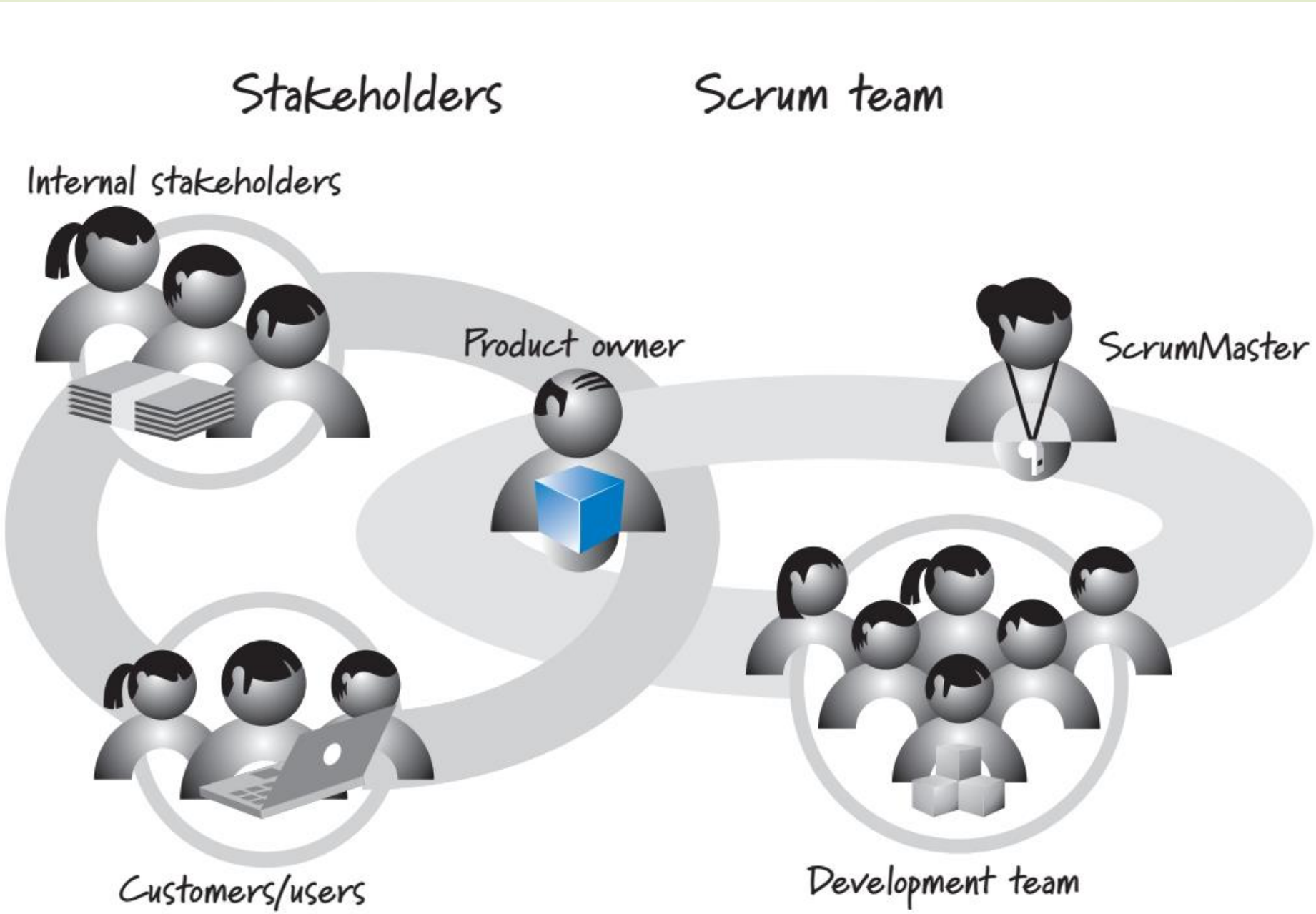
- Self-organizing and cross-functional
  - Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team.
  - Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team.
- Is designed to optimize flexibility, creativity, and productivity.

Team consist of

- A Product Owner,
- The Development Team,
- A Scrum Master

The Product owner(1)

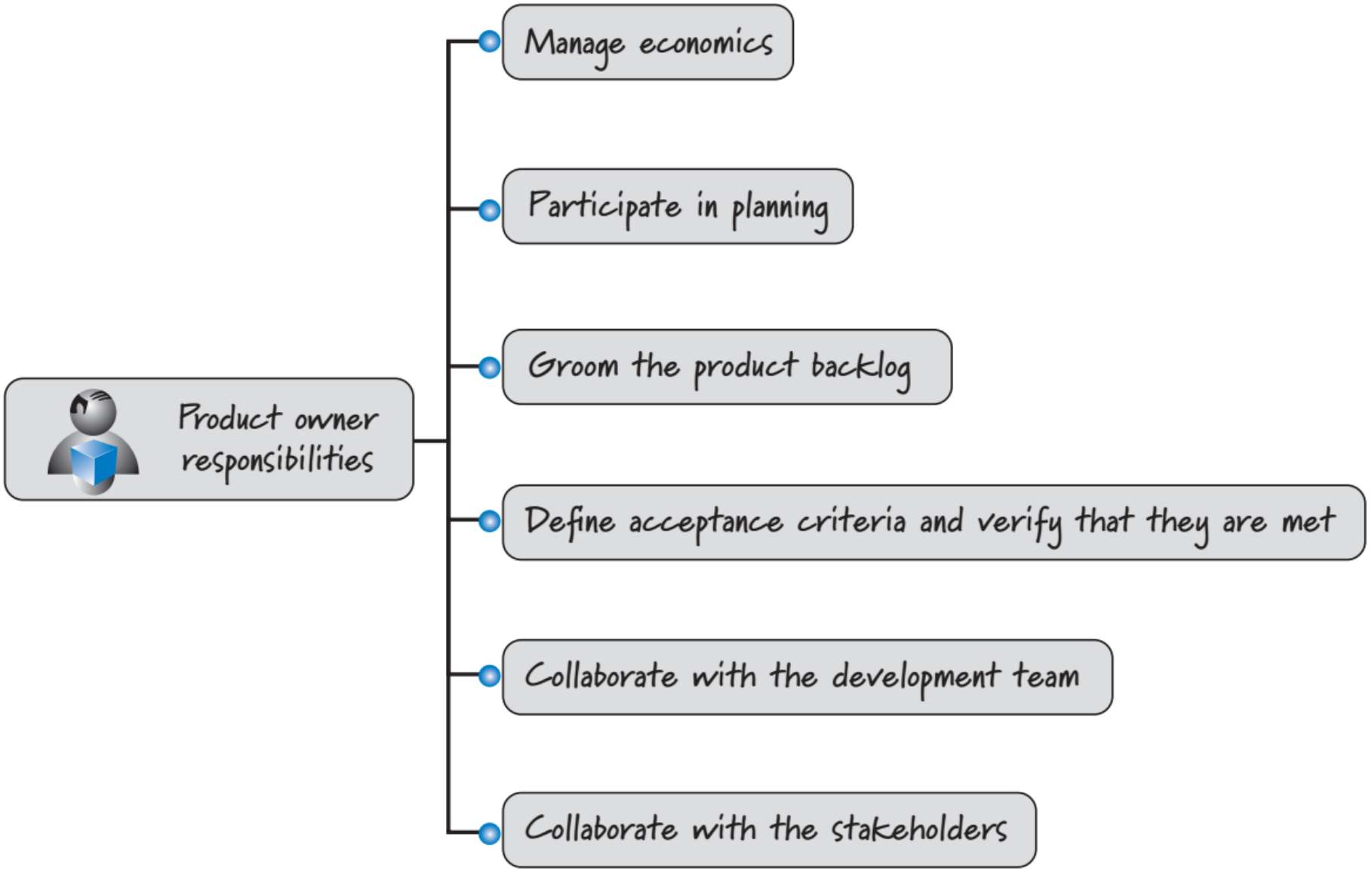
- Responsible for maximizing the value of the product resulting from work of the Development Team.
- The empowered central point of product leadership.
- Needs to look in at least two directions simultaneously.


Stakeholders
Scrum team
Internal stakeholders
Product owner
ScrumMaster
Customers/users
Development team

The Product owner(3)

- Must understand the needs and priorities of the organizational stakeholders, the customers, and the users well enough to act as their voice.
  - In this respect the product owner acts as a product manager, ensuring that the
    right solution is developed.
- Must communicate to the development team what to build and the order in which to build it.
  - Must also ensure that the criteria for accepting features are specified and the tests that verify those criteria are later run to determine whether the features are complete.
  - The product owner doesn't write detail-level tests but ensures that the high-level ones are written so that the team can determine when the product owner will consider the feature complete.
- He/she is part business analyst and part tester.

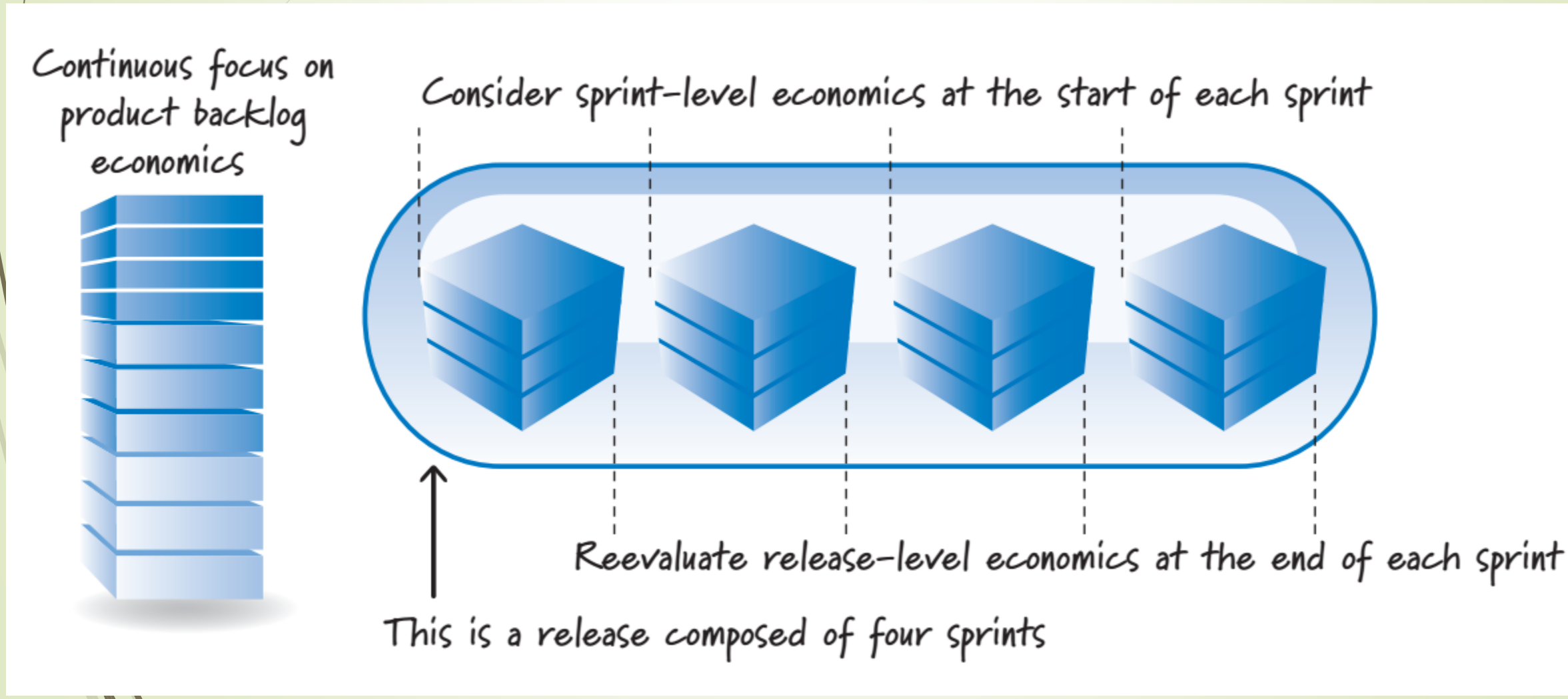
Responsibilities



Product owner responsibility: Manage Economics

- Ensuring that good economic decisions are continuously being made at the release, sprint, and product backlog levels.

Manage Economics



Manage economics: Release-Level Economics(

- Continuously makes trade-offs in scope, date, budget, and quality as a stream of economically important information arrives during product development.
- Trade-offs made at the beginning of a release might no longer be appropriate in the presence of new information that arrives during the release.

Release-Level Economics(2)

- At the end of every sprint the product owner oversees a decision as to
  whether or not to fund the next sprint.
  - If good progress is being made toward the release goal or the next sprint is otherwise economically justified, the next sprint will be funded.
  - If poor progress is being made or the economics don't support additional expenditures, the effort might be canceled.

Release-Level Economics(3)

- Oversee a decision to stop funding further development at the end of a sprint if the product is ready to ship and additional expenditures simply aren't justified.
  - For example: if we planned for a ten-sprint release. After sprint 7, the product owner reviews the remaining product backlog items and concludes that the cost to create those items is greater than the value they deliver. The product owner might conclude that shipping the product early instead of continuing with the original ten-sprint plan is economically more sensible.
- This flexibility to deliver early is enabled by ensuring that the higher-value items at the top of the product backlog are worked on first and the team is completing work each sprint in accordance with a strong definition of done.

Release-Level Economics(4)

- Conclude that we should stop funding at the end of a sprint because core economic properties have changed.
  - For example, what if we are creating a country-specific product and a regulatory agency in that country revises its laws, making it unprofitable or perhaps even illegal for us to sell the product? In cases like these, a product owner might oversee canceling the development effort even if things are otherwise going well.

Sprint-Level Economics

- Ensuring that a good return on investment (ROI) is delivered from each sprint.
- Good product owners treat their organization's money as if it were their own money.
- The product owner knows the cost of the next sprint (the duration and team composition of the sprint are known). With this knowledge the product owner should ask himself at sprint planning, "Would I write a check out of my own bank account equal to the cost of this sprint to get the features that we're planning to build in this sprint?" If the answer is no, a good product owner wouldn't spend the organization's money either.

Product Backlog Economics(1)

- Responsible for prioritizing the product backlog.
- When economic conditions change, the priorities in the product backlog will likely change as well.
  - For example, let's say that at the start of a release the product owner believes a feature is valuable to a large percentage of the target users and the team believes that only a modest effort is required to create it. After a few sprints, however, the team discovers that the feature will require a large effort to complete and is valuable to only a fraction of the target users.
  - Because the cost/benefit ratio of this feature has dramatically changed, the product owner should reprioritize the product backlog to reflect this knowledge—perhaps by dropping the product backlog items associated with the feature.

Product owner responsibility: Participate in Planning(1)

- Is a key participant in the portfolio-, product-, release-, and sprint-planning activities.
- During portfolio planning, the product owner works with internal stakeholders (perhaps an approval committee or governance board) to position the product correctly in the portfolio backlog and to determine when to start and end product development.
- During product planning, the product owner works with the stakeholders to envision the product.

Product owner responsibility: Participate in Planning(2)

- During release planning, the product owner works with the stakeholders and the team to define the content of the next release.

- During sprint planning, the product owner works with the development team to define a sprint goal. He also provides valuable input that enables the development team to select a set of product backlog items that the team can realistically deliver by the end of the sprint.

Product owner responsibility: Groom the Product Backlog(1)

- Oversees the grooming of the product backlog, which includes creating and refining, estimating, and prioritizing product backlog items.
- Doesn't estimate the items (the development team does that) but is available for questions and clarification during estimation.
- Responsible for making sure that the grooming activities take place in a way that promotes the smooth flow of delivered value.

Product owner responsibility: Groom the Product Backlog(2)

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next;
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

Product owner responsibility: Define Acceptance Criteria and Verify That They Are Met(1)

- Responsible for defining the acceptance criteria for each product backlog item.
  - These are the conditions under which the product owner would be satisfied that the functional and nonfunctional requirements have been met.
- Write acceptance tests corresponding to the acceptance criteria, should ensure that these acceptance criteria (and frequently specific acceptance tests) are created before an item is considered at a sprint-planning meeting.
  - Without them, the team would have an incomplete understanding of the item and would not be ready to include it in a sprint.

Product owner responsibility: Define Acceptance Criteria and Verify That They Are Met(2)

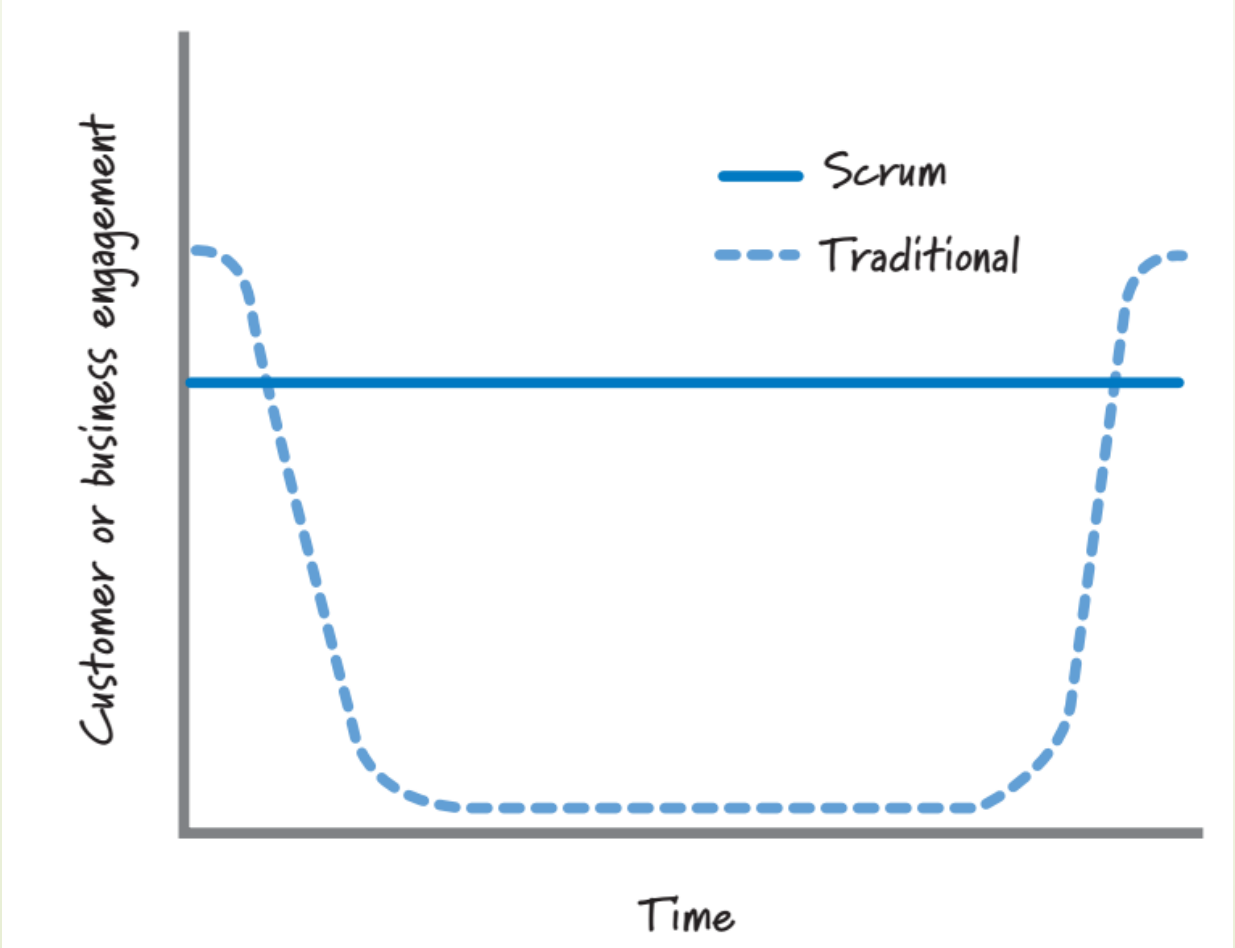
- Responsible for confirming that the acceptance criteria are met.
- The product owner should be the final judge of whether an item meets expectations.
- It is important for the product owner to verify acceptance criteria **during sprint execution** rather than waiting until the sprint review.
- By doing some testing as the features are completed, the product owner can identify mistakes and misunderstandings that the team can fix before the sprint review.

Product owner responsibility: Define Acceptance Criteria and Verify That They Are Met(3)

- Also, because the team is allowed to demonstrate only completed features at the review, the product owner must ensure that acceptance tests are run prior to the review so that the team knows which features actually meet the definition of done.

Product owner responsibility: Collaborate with Development Team(1)

- Must closely collaborate with the development team on a frequent basis.



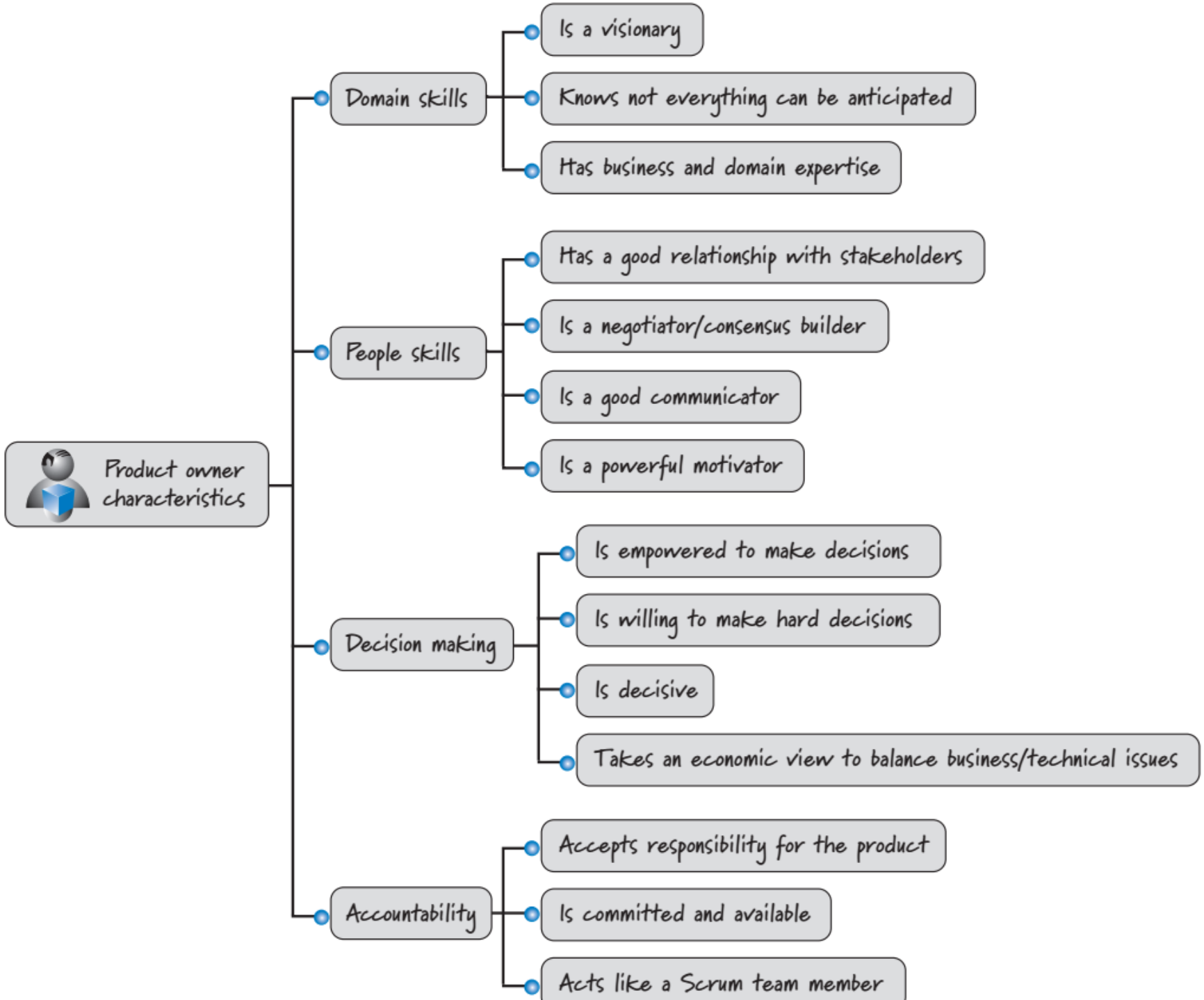
Product owner responsibility: Collaborate with Development Team(2)

- Using Scrum, we build a feature at a time, not a phase at a time. This means that we perform all activities to create a particular feature (design, code, integrate, test) during one sprint.
- Therefore, a constant high level of engagement by the product owner is essential. With such close interaction on short, time-boxed iterations there is far less chance for the product owner and the development team to become disconnected.
- A secondary benefit is that there is no finger-pointing when doing Scrum well!

Product owner responsibility: Collaborate with the Stakeholders

- Is the single voice of the entire stakeholder community, internal and external.
- Internal stakeholders can include business systems owners, executive management, program management, marketing, and sales.
- External stakeholders can include customers, users, partners, regulatory bodies, and others.
- Must work closely with the entire stakeholder community to gather input and synthesize a coherent vision to guide product development.

Product owne characteristics



Domain skills

- The product owner is a visionary who can synthesize a product vision and lead the team to achieve that vision.
- Having a vision doesn't mean that every detail of the vision or the path to achieving the vision is perfectly clear.
- A good product owner knows that not everything can be anticipated up front and is willing to adapt when change is warranted.
- To be effective at vision creation and execution, a product owner must have appropriate business and domain knowledge.

People Skills(1)

- Must be the "voice of the customer," which requires a good relationship with stakeholders. Because there are frequently multiple stakeholders who could have conflicting needs, the product owner must also be a good negotiator and consensus builder.
- The product owner is the linchpin between the stakeholder community
  and the rest of the Scrum team.
- In this position, the product owner needs good communication skills to work with both constituencies and to convey information to each group in the appropriate language.

People Skills(2)

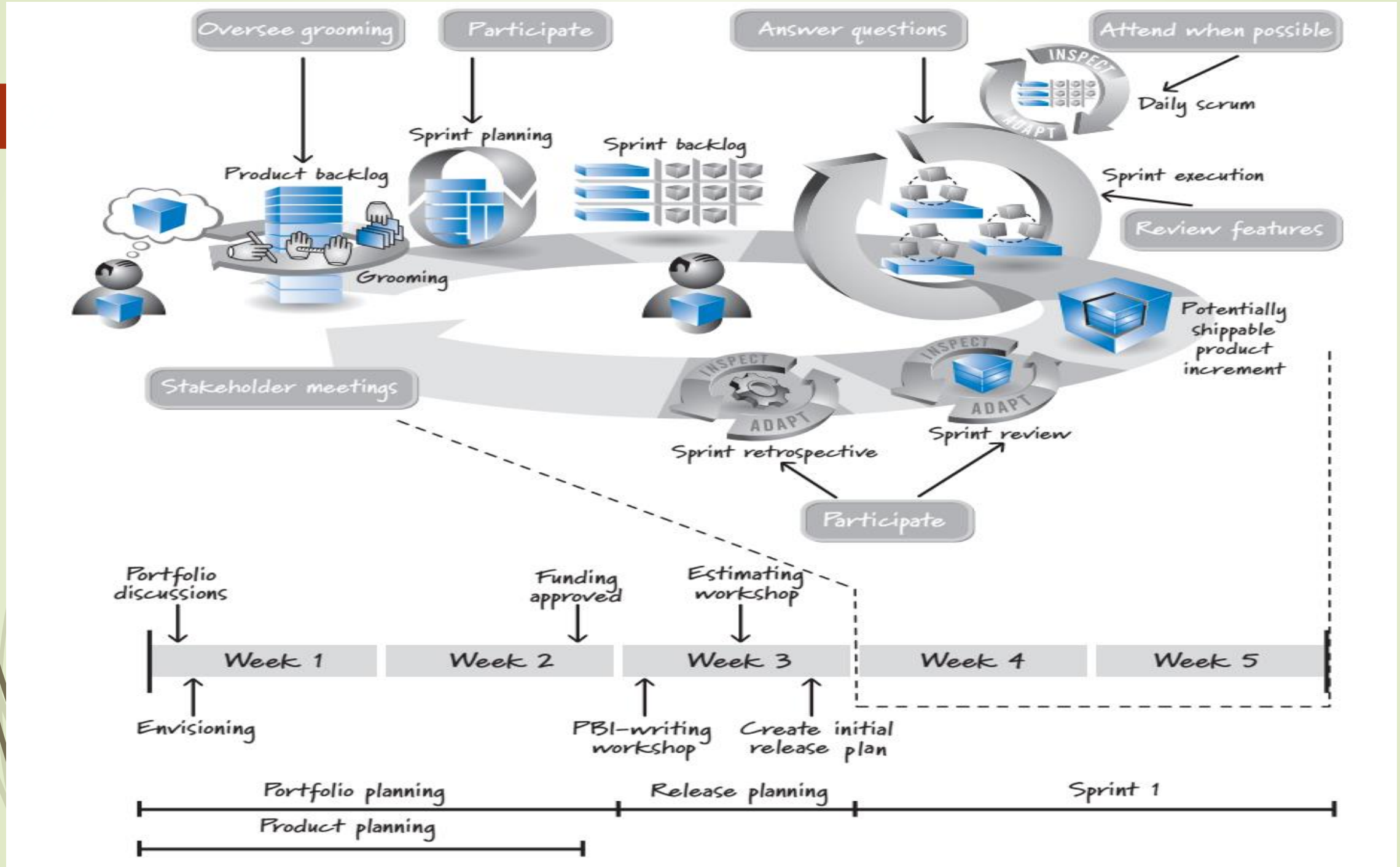
- A good communicator also exhibits the following qualities:
  - is willing to speak up even if doing so goes against the status quo;
  - confident in his ideas;
  - knowledgeable of the subject matter;
  - able to communicate in a simple, concise, and easily understood way;
  - and credible.
- A product owner is also a powerful motivator. When the going gets tough, the product owner can remind people of why they are investing the effort and help people maintain an enthusiastic outlook by reinforcing the business proposition.

Decision Making

- Must be empowered to make decisions.
- The product owner must also be willing to make the hard decisions—usually by trading off constraints like scope, date, and budget.
- These decisions must be made in a timely manner and should not be reversed without good reason.
- In making these decisions, a product owner must maintain the proper balance between business needs and technical realities.

Accountability

- Is accountable for delivering good business results.
- Be sure that the resources are being used in an economically sensible way and must accept responsibility.
- The product owner must be committed and available to both the stakeholders and the rest of the Scrum team.
- Being a product owner is a full-time job;


Oversee grooming
Participate
Answer questions
Attend when possible
Daily scrum
Sprint planning
Sprint backlog
Product backlog
Sprint execution
Review features
Grooming
Potentially shippable product increment
Stakeholder meetings
Sprint retrospective
Sprint review
Participate
Portfolio discussions
Funding approved
Estimating workshop
Week 1
Week 2
Week 3
Week 4
Week 5
Envisioning
PBI-writing workshop
Create initial release plan
Portfolio planning
Release planning
Sprint 1
Product planning

A Day in the Life of the product owner(2)

- During weeks 1 and 2 the product owner is engaged in both portfolio planning and product planning.
  - As part of portfolio planning, the product owner might work with the portfolio manager or the governance board to discuss portfolio expectations that could influence the new-product planning. Those discussions provide input to product planning, where the product owner, working with appropriate stakeholders and others, will perform the envisioning of the new product.
  - At the completion of product planning, the proposed product will be submitted to portfolio planning, where it is subjected to the organization's economic filter to determine if development will be funded and when work can begin.

A Day in the Life of the product owner(3)

- In week 3 the product owner is engaged in initial release planning. This typically involves a PBI-writing (story-writing) workshop that includes internal stakeholders, the development team members, and possibly external stakeholders to generate a high-level product backlog that can be used during release planning. The development team members should be available to participate because funding has already been approved.
- Next, the product owner facilitates an initial release-planning session. Because some number of product backlog items have already been estimated, the focus of this release-planning activity is on prioritizing the product backlog and balancing the constraints of scope, schedule, and budget.

A Day in the Life of the product owner(4)

- Scrum team performs the first sprint shows a two-week sprint during weeks 4 and 5). At the start of the sprint the product owner oversees the sprint-planning activity.
- During sprint execution, the product owner tries to attend the team's daily scrums; this may not always be possible, but it is a good practice.
- During the daily scrum, the product owner listens to better understand how the current sprint is progressing and to identify opportunities to assist the development team.

A Day in the Life of the product owner(5)

- Perhaps a team member mentions he is a little fuzzy on the specifics of a product backlog item and needs some clarification before he can complete his current task. If it is a quick clarification, the product owner might offer it up during the daily scrum. If the answer requires anything other than a few-second response, the product owner should say, "I'd be happy to stick around after the daily scrum and discuss it with you."
- The product owner must also be available (typically every day) to answer questions and to test features as they become reviewable. If the product owner knows he can't be available every day to perform these responsibilities, he must delegate them to an appropriate person so that the development team will not be blocked.

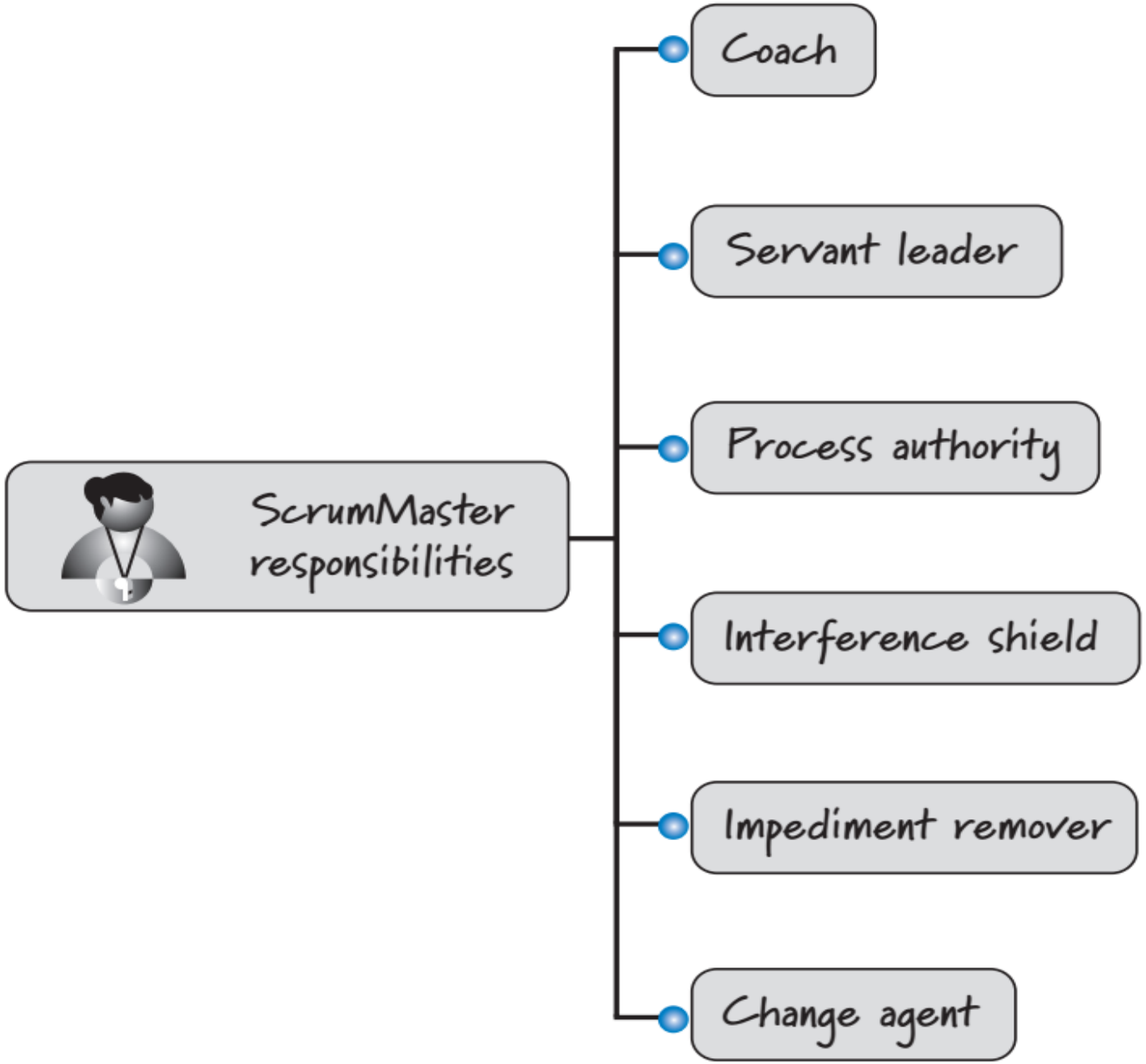
A Day in the Life of the product owner(6)

- Also during sprint execution the product owner meets with both the internal and external stakeholders to ensure that priorities for the upcoming sprint are correct and to secure valuable user input that will affect the features chosen for future sprints.
- The product owner also performs frequent product backlog grooming, which includes writing new product backlog items and refining existing items, then working with the team to get them estimated, and the stakeholders and team to get them prioritized.
- At the end of the sprint, the product owner participates in the two end-of-sprint inspect-and-adapt activities: the sprint review and the sprint retrospective.
- After these are completed, the sprint cycle repeats and the product owner participates in the next sprint-planning activity.

Scrum Master

- Is one of the three roles that constitute every Scrum team.
- While the product owner is focused on building the right product and the development team is focused on building the product right, the **ScrumMaster is focused on helping everyone understand and embrace the Scrum values, principles, and practices**.
- Acts as a coach to both the development team and the product owner.
- Provides process leadership, helping the Scrum team and the rest of the organization develop their own high-performance, organization-specific Scrum approach.

Principal Responsibilities



Principle responsibilities: Coach(1)

- Is the agile coach for the Scrum team
- Can remove barriers between the roles and enable the product owner to directly drive development.
- Observes how the team is using Scrum and does anything possible to help it get to the next level of performance.
- When problems arise that the team can and should be able to solve, “I’m here to help you solve your own problems.” If the problem is an impediment that the team can’t resolve, the ScrumMaster takes ownership of getting it resolved.

Principle responsibilities: Coach(2)

- Coaches a new product owner by helping him understand and perform his product owner responsibilities.
- Provides him with ongoing assistance for activities such as grooming the product backlog.
- Help the owner maximize business outcomes using Scrum, manage expectations, make sure the owner is providing the team with what it needs, and listen to the owner's complaints and requests for change and translate those into actionable improvements for the team.

Principle responsibilities: Servant leader

- Is first and foremost a servant to the Scrum team, ensuring that its highest-priority needs are being met.
- A servant leader would never ask, “So, what are you going to do for me today?” Instead, a servant leader asks, “So, what can I do today to help you and the team be more effective?”

Principle responsibilities: Process Authority

- Is empowered to ensure that the Scrum team enacts and adheres to the Scrum values, principles, and practices along with the Scrum team's specific approaches.
- Continuously helps the Scrum team improve the process, whenever possible, to maximize delivered business value.
- Authority in this context is not the same type of authority that a functional manager or project manager would have.
  - For example, the ScrumMaster doesn't hire and fire and cannot dictate to the team what tasks it should do or how to do them.
- Is not responsible for making sure the work gets done. Instead, he/she helps the team define and adhere to its own process for making sure the work gets done.

Principle responsibilities: Interference Shield

- Protects the development team from outside interference so that it can remain focused on delivering business value every sprint.
- Interference can come from any number of sources, from managers who want to redirect team members in the middle of a sprint, to issues originating from other teams.
- No matter what the source of the interference, the ScrumMaster acts as an interceptor so that the team can focus on delivering value.

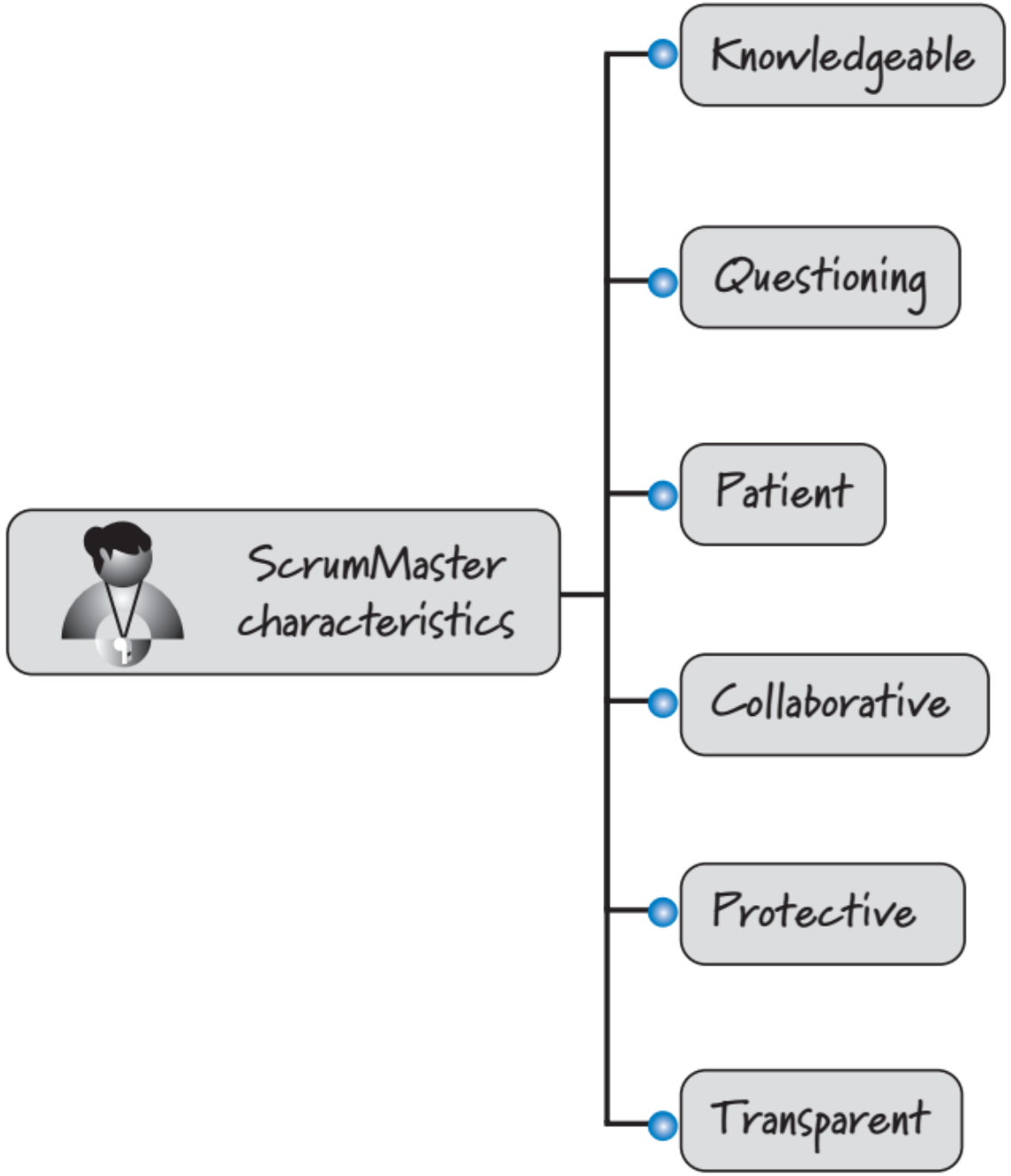
Principle responsibilities: Impediment Remover

- Takes responsibility for removing impediments that inhibit the team's productivity (when the team members themselves cannot reasonably remove them).

Principle responsibilities: Change Agent

- Must help change.
- Scrum can be very disruptive to the status quo; the change that is required to be successful with Scrum can be difficult.
- The ScrumMaster helps others understand the need for change, the impacts of Scrum outside of the Scrum team, and the broad-reaching benefits Scrum can help achieve.
- The ScrumMaster also ensures that effective change is occurring at all levels of the organization, enabling not only short-term success but, more importantly, the long-term benefits from using Scrum.

Skills


ScrumMaster characteristics
Knowledgeable
Questioning
Patient
Collaborative
Protective
Transparent

Knowledgeable

- To be an effective process coach, the ScrumMaster must be very knowledgeable about Scrum.
- Should also understand the technical issues the team needs to address and technologies the team will use to create solutions.
- Doesn't need to have tech-lead- or dev-lead-level knowledge, but reasonable technical knowledge is an asset.
- Doesn't need to be an expert in the business domain (the product owner does), but again, working knowledge of the business domain is very helpful.

Questioning

- Use their coaching skills in conjunction with their process, technical, and business knowledge to ask great questions. They engage in intentional inquiry, asking the kinds of questions that make people stop and say, "Hmm. I never thought about that. Now that you ask that question, it makes me think there might be another way to go."
- Great ScrumMasters almost never directly answer a question but instead

  reflexively answer with their own question—not an annoying question, or a question for the sake of asking a question, but rather a thoughtful, deep, probing question—thereby helping people realize that they have the insight to find their own answers a form of Socratic questioning.

Patient

- Because ScrumMasters prefer not to give out answers, they need to be patient, giving teams time to arrive at appropriate answers on their own. At times it is hard for me to be a ScrumMaster because I see the issue the team is dealing with and I "know" the answer. Well, at least I *think* I know the answer! It is arrogant for me (or any ScrumMaster) to believe that I am smarter than the collective intelligence of the team.
- So, at times I just have to bite my tongue and be patient, letting the team work out the solution, periodically asking probing questions to help guide things along.

Collaborative

- Must have excellent collaboration skills to work with the product owner, development team, and all the other parties, even those who might not be directly involved with Scrum.
- Also, as the process coach, the ScrumMaster is always looking for opportunities to help the Scrum team members achieve an enviable level of intra-team collaboration.

Protective

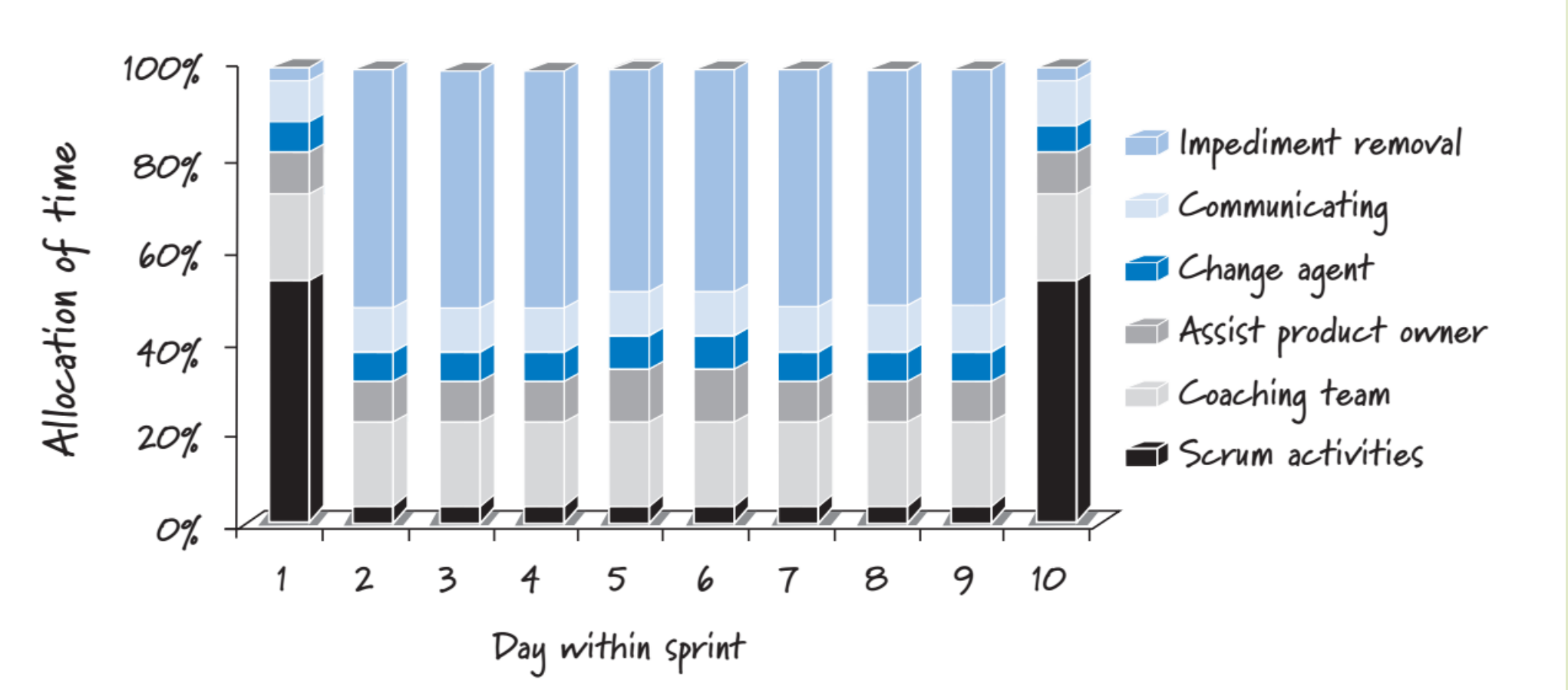
- Should be very protective of the team. The common analogy is that the ScrumMaster acts like a sheepdog, guarding the flock from wolves that might try to attack. In our context wolves could be organizational impediments or people with differing agendas.
- Is adept at ensuring the protection of the team within the greater context of making economically sound business decisions. With acute sensitivity toward both team protection and business needs, the ScrumMaster helps the Scrum team achieve a healthy balance.
- The ScrumMaster also helps team members who begin to wander away from the flock. When things get difficult, it is easy for people to fall back on familiar, non-agile approaches. In this case it is the ScrumMaster's job to help shepherd straying team members, helping them overcome their difficulties by reinforcing how to use Scrum more effectively.

Transparent

- Is transparent in all forms of communication. When working with team members, there is no room for hidden agendas; what you see and hear from the ScrumMaster must be what you get.
- People expect nothing less of a servant leader.
- The ScrumMaster also promotes transparent communication outside of the Scrum team.
- Without transparent access to information it is difficult for the organization to inspect and adapt to achieve its desired business results from using Scrum.

A day in the life

- How much time the ScrumMaster of **a newly formed team** might spend doing each of the activities throughout a sprint.



Scrum activities

- ScrumMaster spends time each day organizing and facilitating the Scrum activities, including sprint planning, sprint execution, sprint reviews, sprint retrospectives, and daily scrums. This includes setting up the activities, overseeing their execution, and enabling the rest of the Scrum team to perform at a level where high-value results are achieved.

Coaching

- The ScrumMaster also spends time each day coaching the team members to help them improve their use of Scrum and technical practices.
- Also perform refresher training.
- Also, some amount of each day is dedicated to communicating (for example, updating sprint and release burndown or burnup charts, or discussions with non-Scrum-team members).

Assist product owner

- ScrumMaster spends some time working with the product owner on product backlog grooming activities (for example, writing and prioritizing new product backlog items).
- Also works with the product owner to ensure that economically viable trade-offs are being made regarding important variables such as feature, date, budget, and quality.

Change agent

- The ScrumMaster also spends time acting as a change agent to help the organization better embrace Scrum throughout the value chain (from sales, marketing and so on).

Impediment remover

- The ScrumMaster spends a variable amount of time each day removing impediments. She might budget a fixed amount of time each day specifically for impediment removal.
- Of course, impediments can appear at any time, and they might be large and time-sensitive, so the ScrumMaster might need to dynamically reallocate time from other activities to address them.
- Most teams and organizations that are new to Scrum will have many impediments when starting out and tend to focus on the ones that are obvious and somewhat easy to remove. That doesn't mean, however, that all impediments will be easily dispatched.
- In fact, the next level of impediments is often much more difficult and time-consuming to address. Impediment removal is a big variable in the ScrumMaster's day;

Development Team

- Traditional software development approaches define various job types, such as architect, programmer, tester, database administrator, UI designer, and so on.
- Scrum defines the role of development team, which is simply a cross-functional collection of these types of people.
- In particular, the development team is one of the three roles on every Scrum team. The development team's members, collectively, have the skills required to deliver the business value requested by the product owner.

Role-specific team(1)

- Many organizations are accustomed to intentionally splitting different job roles into specialized, role-specific teams. These organizations might have one team of designers, one of developers, and another of testers. These teams hand off work to one another when it is complete and more or less function independently of each other.
- In Scrum, the development team must do all of the work to produce one or more vertical slices of working product functionality each sprint, including the design, development, integration, and testing of that functionality. Thus, we need a team that is skilled at all of those tasks.

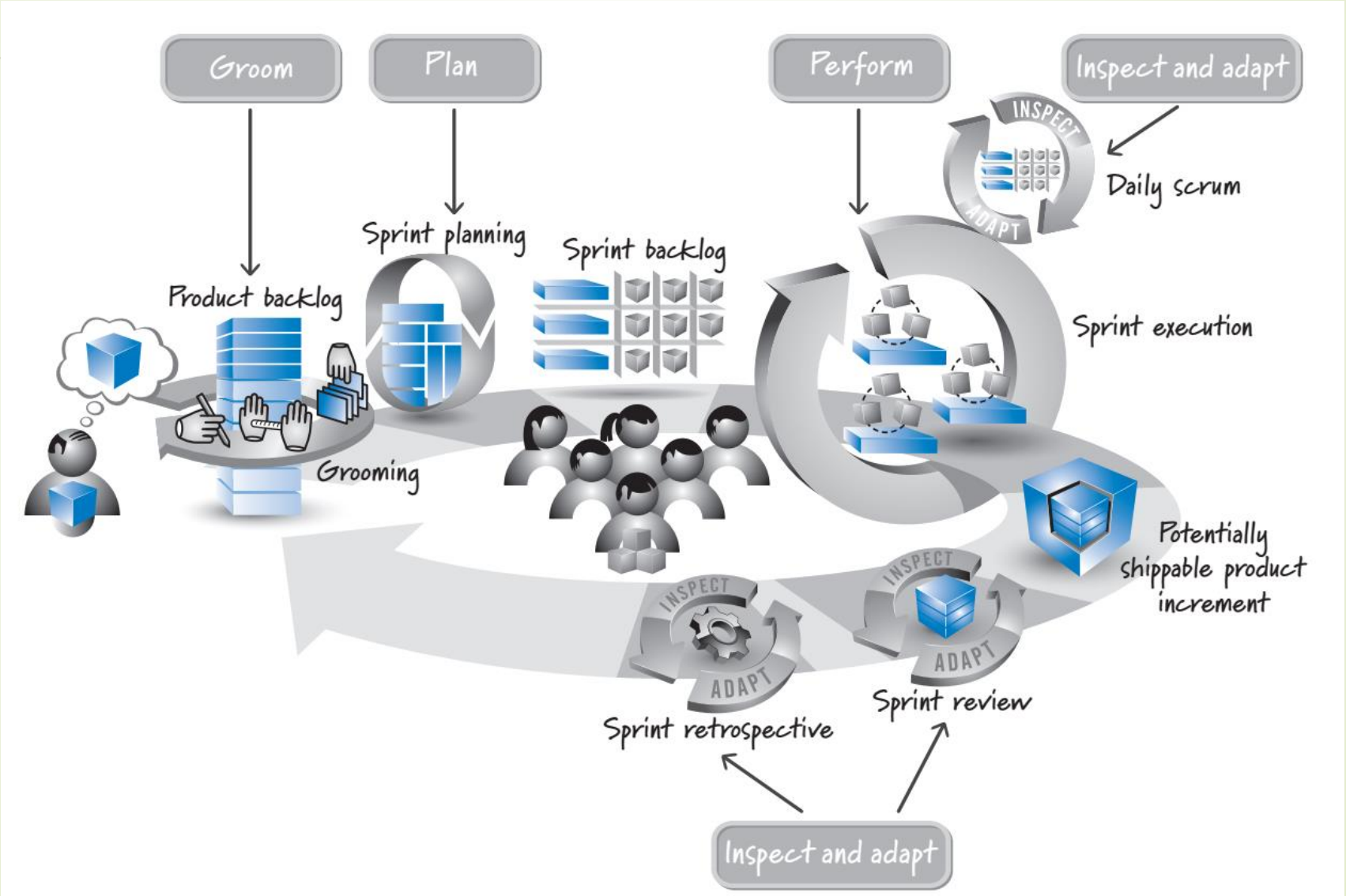
Role-specific team(2)

- Some organizations try to maintain a separate testing or QA team while doing Scrum. Now, I admit there are times when having a separate team that focuses specifically on testing might be necessary—for example, a regulatory requirement might be that a separate team perform a particular type of testing.
- However, most of the time there is no such need. Testing should be fully interwoven into the work that takes place during every sprint. Therefore, the development team doing the work during that sprint should do the testing.

Role-specific team(3)

- Whenever you can, you should create cross-functional teams. Parceling the work out to different role-specific teams is suspect and is likely a serious impediment to the successful use of Scrum.
- Make sure you have a real need (besides habit) for keeping any role-specific teams.

Principal Responsibilities



Principal Responsibilities: Perform Sprint Execution

- During sprint execution, development team members perform the hands-on, creative work of designing, building, integrating, and testing product backlog items into increments of potentially shippable functionality.
- To do this, they self-organize and collectively decide how to plan, manage, carry out, and communicate work. The development team spends a majority of its time performing sprint execution.

Principal Responsibilities: Inspect and Adapt Each Day

- Each development team member is expected to participate in each daily scrum, during which the team members collectively inspect progress toward the sprint goal and adapt the plan for the current day's work.
- If some team members do not participate, the team can miss pieces of the big picture and may fail to achieve its sprint goal.

Principal Responsibilities: Groom the Product Backlog

- Part of each sprint must be spent preparing for the next. A large part of that work focuses on product backlog grooming, which includes creating and refining, estimating, and prioritizing product backlog items.
- The development team should allocate up to 10% of its available capacity every sprint to assist the product owner with these activities.

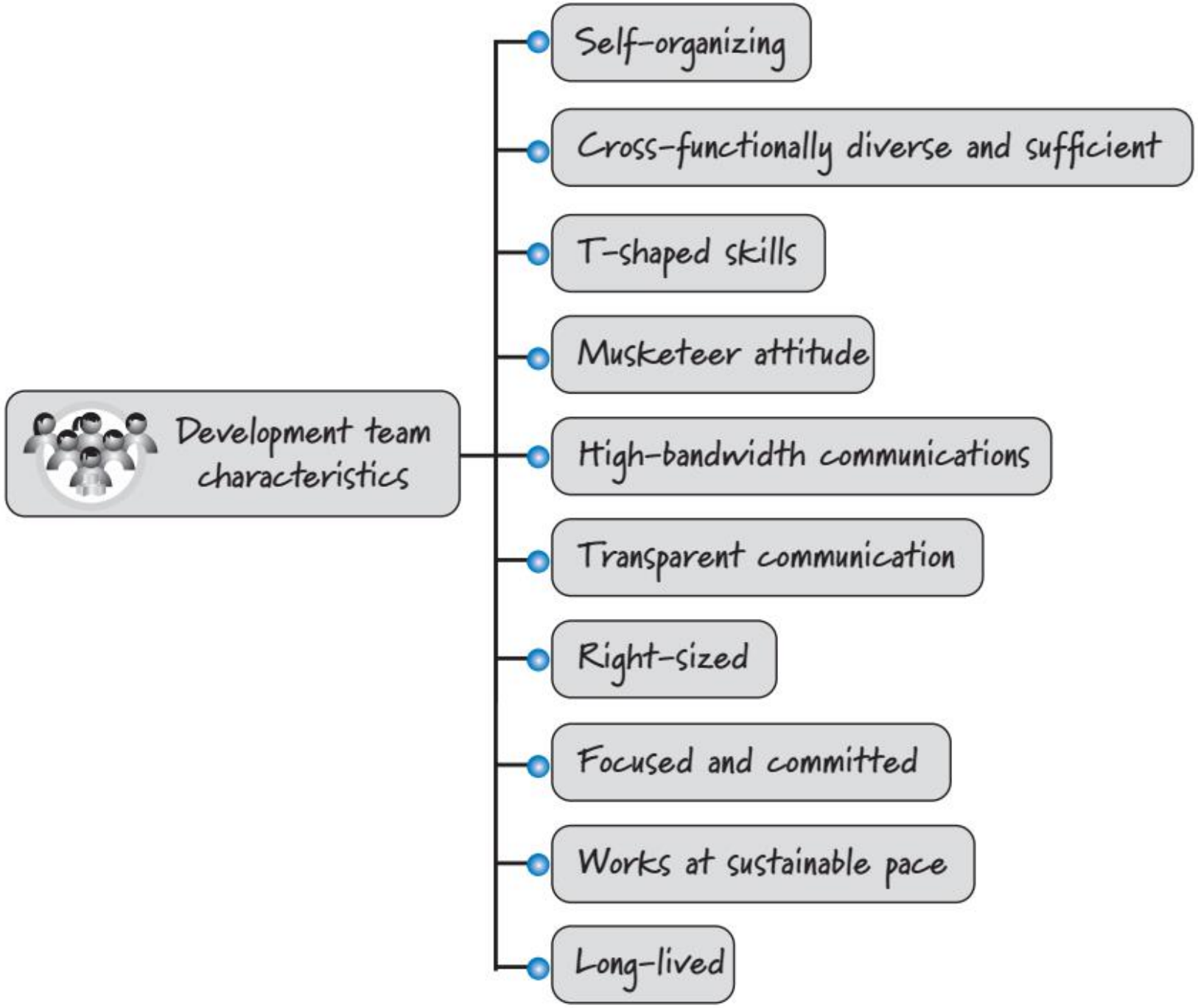
Principal Responsibilities: Plan the Sprint

- At the beginning of each sprint, the development team participates in sprint planning. In collaboration with the product owner and with facilitation from the ScrumMaster, the development team helps to establish the goal for the next sprint.
- The team then determines which high-priority subset of product backlog items to build to achieve that goal.
- Notice that planning happens iteratively. Rather than focusing on a very large, uncertain, and overly detailed plan at the start of a development effort, the team makes a series of smaller, more certain, and more detailed plans just in time at the beginning of each sprint.

Principal Responsibilities: Inspect and Adapt the Product and Process

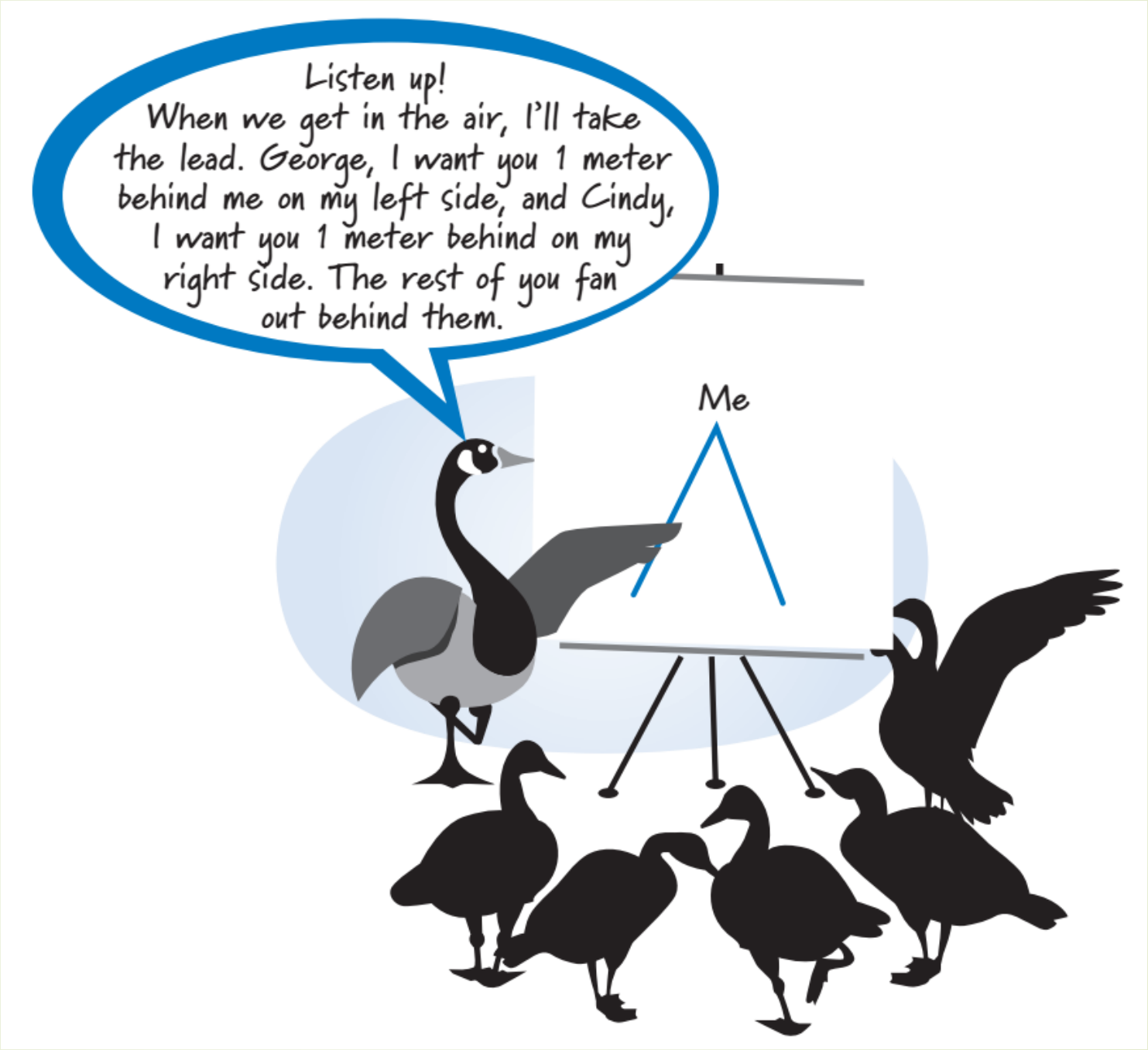
- At the end of each sprint, the development team participates in the two inspect-and adapt activities: sprint review and sprint retrospective.
- The sprint review is where the development team, product owner, ScrumMaster, stakeholders, sponsors, customers, and interested members of other teams review the just-completed features of the current sprint and discuss how to best move forward.
- The sprint retrospective is where the Scrum team inspects and adapts its Scrum process and technical practices to improve how it uses Scrum to deliver business value.

Development team characteristics



Self-organizing

- Team members self-organize to determine the best way to accomplish the sprint goal.
- There is no project manager or other manager who tells the team how to do its work (and a ScrumMaster should never presume to).
- Self-organization is a bottom-up, emergent property of the system—there is no external dominating force applying traditional top-down, command-and-control management.


Listen up!
When we get in the air, I'll take the lead. George, I want you 1 meter behind me on my left side, and Cindy, I want you 1 meter behind on my right side. The rest of you fan out behind them.
Me

Self-organizing

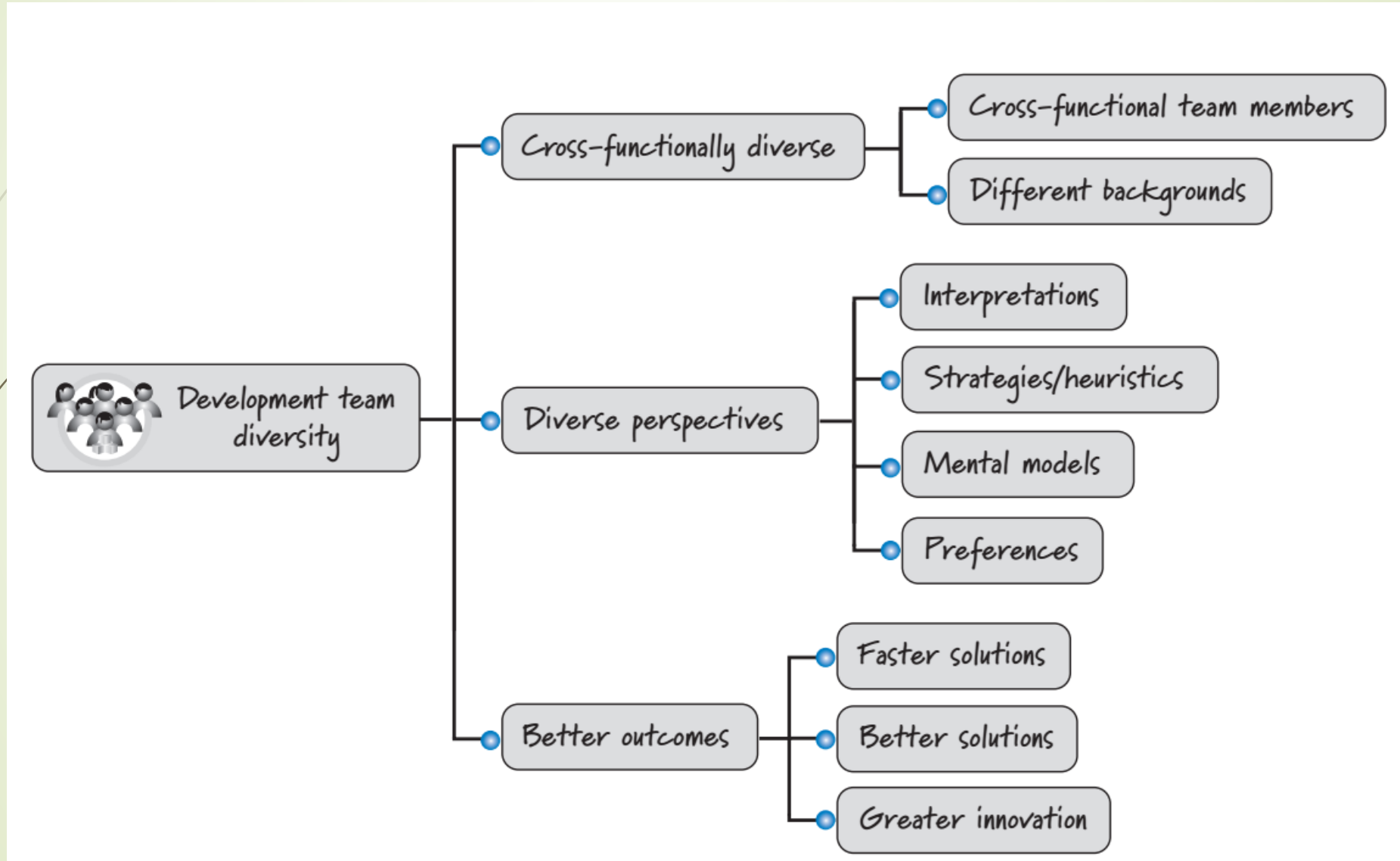
- Like the flocking birds, a development team has no top-down command-and control authority that tells the team how to do its work.
- Instead, a cross-functionally diverse team of people organize themselves in the most appropriate way to get work done.


• Avoid crowding neighbors
• Fly in same general direction as neighbors
• Fly toward average position of neighbors

Cross-Functionally Diverse and Sufficient

- Development team members should be cross-functionally diverse;
- Collectively they should possess the necessary and sufficient set of skills to get the job done. A well-formed team can take an item off of the product backlog and produce a good-quality, working feature that meets the Scrum team's definition of done.
- Teams composed solely of people with the same skills can at most do part of the job. As a result, silo teams end up handing off work products to other silo teams.
- On the other, creating diverse teams doesn't prevent us from having multiple team members who might be highly skilled in the same discipline.

Team Diversity



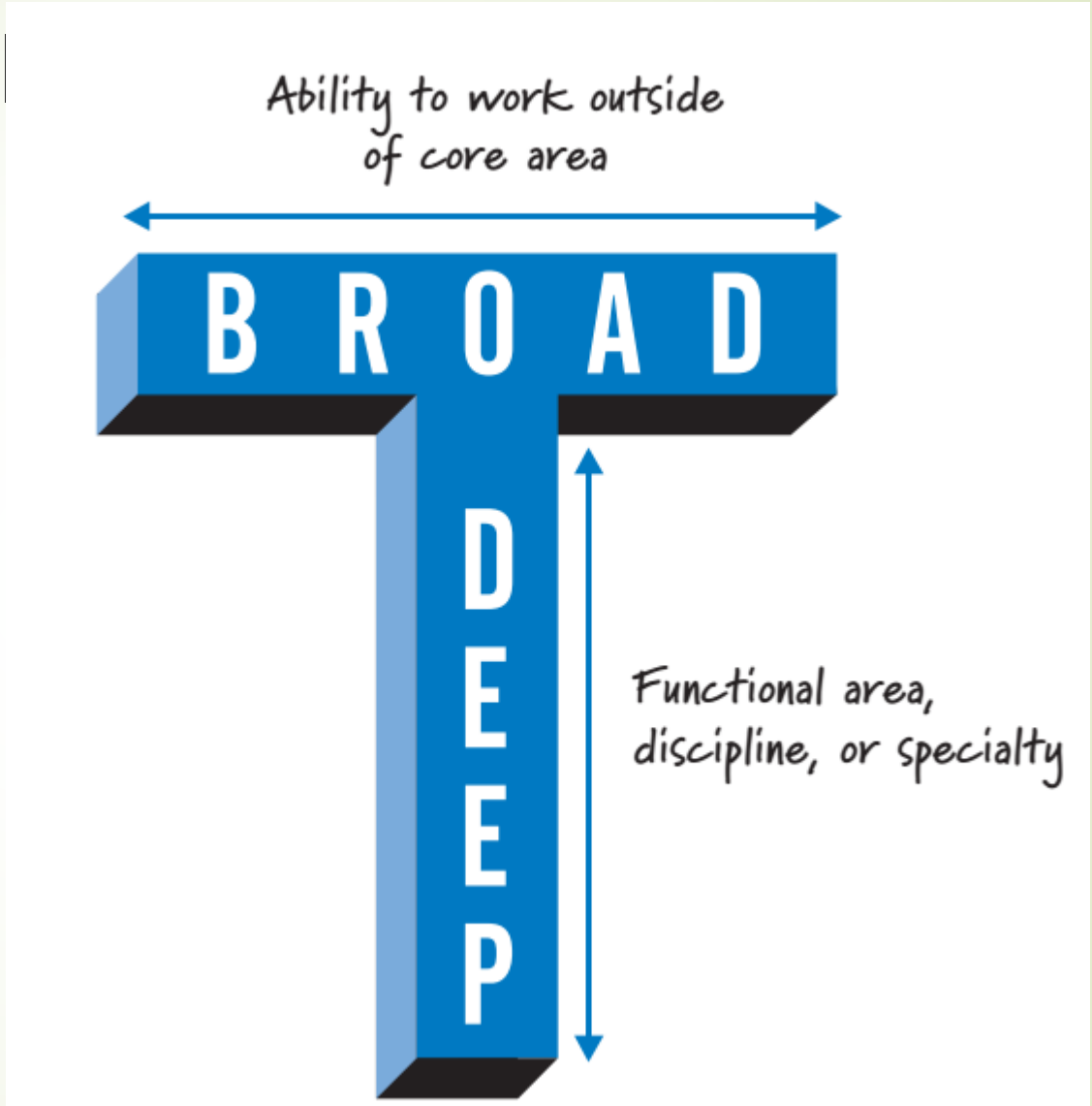
Cross-Functionally Diverse and Sufficient

- A cross-functionally diverse team has members from different backgrounds.
- Each team member brings a set of cognitive tools for problem solving; these tools can involve different interpretations (of the same data), different strategies (or heuristics) for solving problems, different mental models of how things work, and different preferences for both approaches and solutions.
  - This kind of diversity typically leads to better outcomes in terms of faster solutions, higher-quality deliverables, and greater innovation, all of which translate into greater economic value.
- We should also strive for team diversity by having a good mix of senior- and junior-level personnel on the same team. A good mix promotes a healthy, collaborative learning environment.

T-Shaped Skill

- Flexible development teams are composed of members with T-shaped skills.
- Mean that a team member has deep skills in her preferred functional area, discipline, or specialty.
- She isn't as good an specialist as those who specialize in those areas, but she can help out with testing or documentation if that's where the team is experiencing a bottleneck and needs to swarm people to get the job done. In this respect she has broad skills that allow her to work outside her core area.

T-Shaped Skil



T-shaped skill

- It's unrealistic to believe that every person on a team could work on every task.
- Managers should focus on forming teams that have the best set of T-shaped skills that are possible with available personnel.
- However, it might not be possible to get exactly the desired team skill set from the get-go, so the desired skill set could evolve over time as the needs of the product development effort evolve.
- Therefore, it is critical to have an environment where people are constantly learning and adding to their skill sets, whether those include domain knowledge, technical knowledge, thinking skills, or other capabilities.
- Management needs to support team members with time to learn and experiment.

T-shaped skill

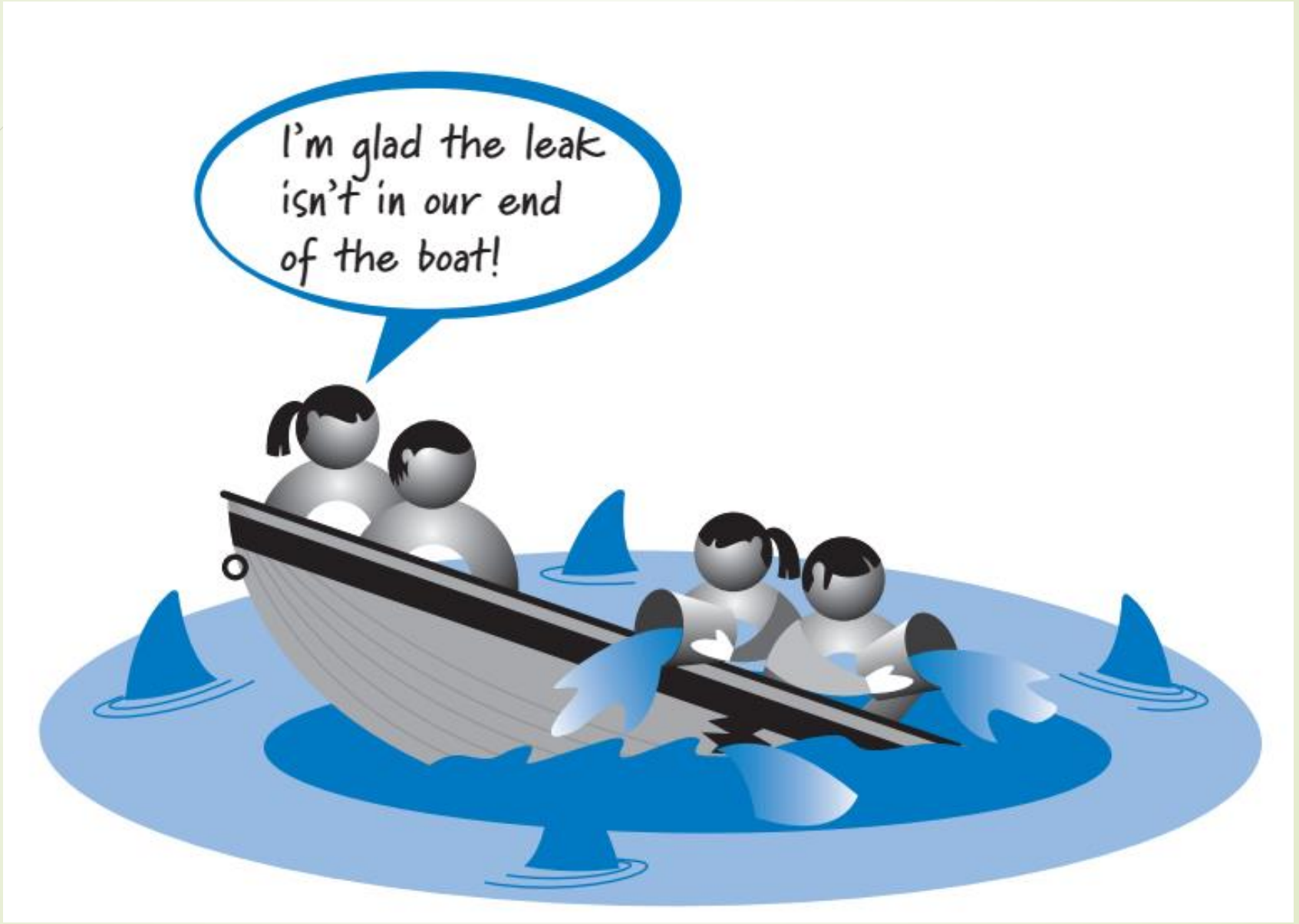
- If we have an expert UX designer and we really don't want her doing anything but critical UX design work.
- We need her skills on the team, but we'll be able to fill only about 10% of her
time with team-related work.
- In these cases, an obvious solution is to divide her time among multiple teams.
- She would be far too fractured if she divided her time in 10% increments to many teams at the same time. She would soon become a bottleneck.
- our goal shouldn't be to keep people100% utilized. Instead, we should be more concerned about the idle work that occurs when we rely too much on an over utilized resource.
- So, we might allocate her as a specialist to a reasonable number of products, but not so many that she is the cause of baton dropping.

T-shaped skill

- Alternatively, because our goal is to achieve good flow with team members who have broad T-shaped skills, we should encourage her to help other team members acquire reasonable UX design knowledge so that we no longer need to rely so heavily on specialists.
- Our goal is to form a team with members who have the proper skills to cover the core specialty areas and in aggregate have some overlap in skills to provide additional flexibility.
- To meet this goal, many team members should have T-shaped skills, but we still might have some specialists in the mix.

Musketeer Attitude

- Members of the development team (and the Scrum team as a whole!) need to have the same attitude as the Three Musketeers—"All for one and one for all."
- This Musketeer attitude reinforces the point that the team members collectively own the responsibility of getting the job done.
- They win as a team or they fail as a team.
- In a well-functioning Scrum team, I would never expect anyone to say, "I got my part done. You didn't get your part done. Therefore we failed." This attitude misses the point that team members are all in the same boat together.


I'm glad the leak isn't in our end of the boat!

Musketeer Attitude

- Team members must appreciate that they must work together to meet their commitments, because if they fail, it's going to be everybody's problem in the end.
- It is critical to achieving shared success.
- Having team members with T-shaped skills encourages this attitude and makes it practical because people are capable of working on more than one type of task.
- On these teams I don't expect to hear a person who is capable of doing the work say, "That's not my job." However, because it is not always possible for a person to do every job, I might hear someone say, "I'm not capable of doing that job." In this case the team might choose to have the person without the skills apprentice with a person who has the skills so that in the future the team will have greater aggregate capabilities.

Musketeer Attitude

- Even if skills limitations prevent people from working cross-functionally, team members can still organize their work to ensure a good flow through the sprint so that no one team member is overburdened.
- With a Musketeer attitude, no one is just "along for the ride." Each team member is responsible for making sure she is fully engaged at all times.
- Frequently this will mean speaking up and engaging in activities outside one's specialty to add to the diversity of the discussion.
- For example, although a team member's specialty might be testing, if she thinks there is a problem in the design the team is coming up with for a given feature, it's her duty to speak up, rather than say, "Not my job; they know better than I do anyway."

High-Bandwidth (fast and efficient with minimal overhead**) Communications**

- Development team members need to communicate with one another, as well as with the product owner and ScrumMaster, in a high-bandwidth manner, where valuable information is exchanged quickly and efficiently with minimal overhead.
- High-bandwidth communications increase both the frequency and quality of information sharing.
- As a result, the Scrum team has more frequent opportunities to inspect and adapt, leading to better and faster decision making.
- Because the economic value of information is time-sensitive, accelerating the rate of information sharing allows the team to maximize its value.
- By quickly exploiting emergent opportunities and recognizing wasteful situations, the team can avoid expending more resources by going down the wrong path.

Ways to achieve high-bandwidth communications

- Face-to-face communication is a preferred approach.
- Certainly, team members who are physically separated or primarily use non-interactive communication (such as documents) are at a disadvantage to collocated team members engaged in real-time, face-to-face collaboration.
- For distributed teams, a certain level of technology support can help improve communication bandwidth.

High-Bandwidth Communications

- Having teams composed of cross-functional team members is a critical step toward achieving high-bandwidth communications. Such teams have more streamlined communication channels simply because they have easy access to the people needed to get the job done.
- Also, such cross-functionally diverse teams are far less likely to have formal handoffs (which usually take the form of written documents) from one team to another. With everyone on the same team, the frequency and formality of handoffs are reduced, which improves communication speed.
- We should also reduce time spent on ceremonies where team members perform a process that adds little or no value.
- We need to identify and eliminate the impediments to improve overall team communication performance.

High-Bandwidth Communications

- Having small teams also improves bandwidth. Communication channels within a team do not scale linearly with the number of team members but instead increase by the square of the number of people on the team according to the formula $N(N - 1)/2$.
- So, if there are 5 people on the team, there are 10 channels of communication. If there are 10 people on the team, there are 45 channels of communications.
- More people means more communication overhead and therefore lower bandwidth.

Transparent Communication

- Provides a clear understanding of what is actually happening to avoid surprises and help build trust among the team members.
- People were frequently surprised ("astonished") to later learn that his communications were intentionally opaque and designed to mislead.
- This resulted in other team members not trusting this individual, which in turn impeded the team's ability to self-organize and meet its sprint goals.

Right-sized

- Scrum favors small teams.
- The general rule is that having five to nine people on the team is best.
- Small teams tend to be the most efficient.

Handful of reasons to keep teams small

- There is less social loafing—people exerting less effort because they believe
  that others will pick up the slack.
- Constructive interaction is more likely to occur on a small team.
- Less time is spent coordinating efforts.
- No one can fade into the background.
- Small teams are more satisfying to their members.
- Harmful overspecialization is less likely to occur.

Right-sized

- Just because Scrum favors small teams doesn't mean we can't use Scrum on
larger development efforts.
- However, rather than having one large Scrum team with, we would instead have four or more Scrum teams, each with a development team of 9 or fewer people.
- A Scrum project scales not by having a larger development team but by having multiple Scrum teams.
- Multiple Scrum teams can coordinate with each other in a variety of ways. One common approach is known as the scrum of scrums, where members of each Scrum team come together to perform a higher-level equivalent of the daily scrum.

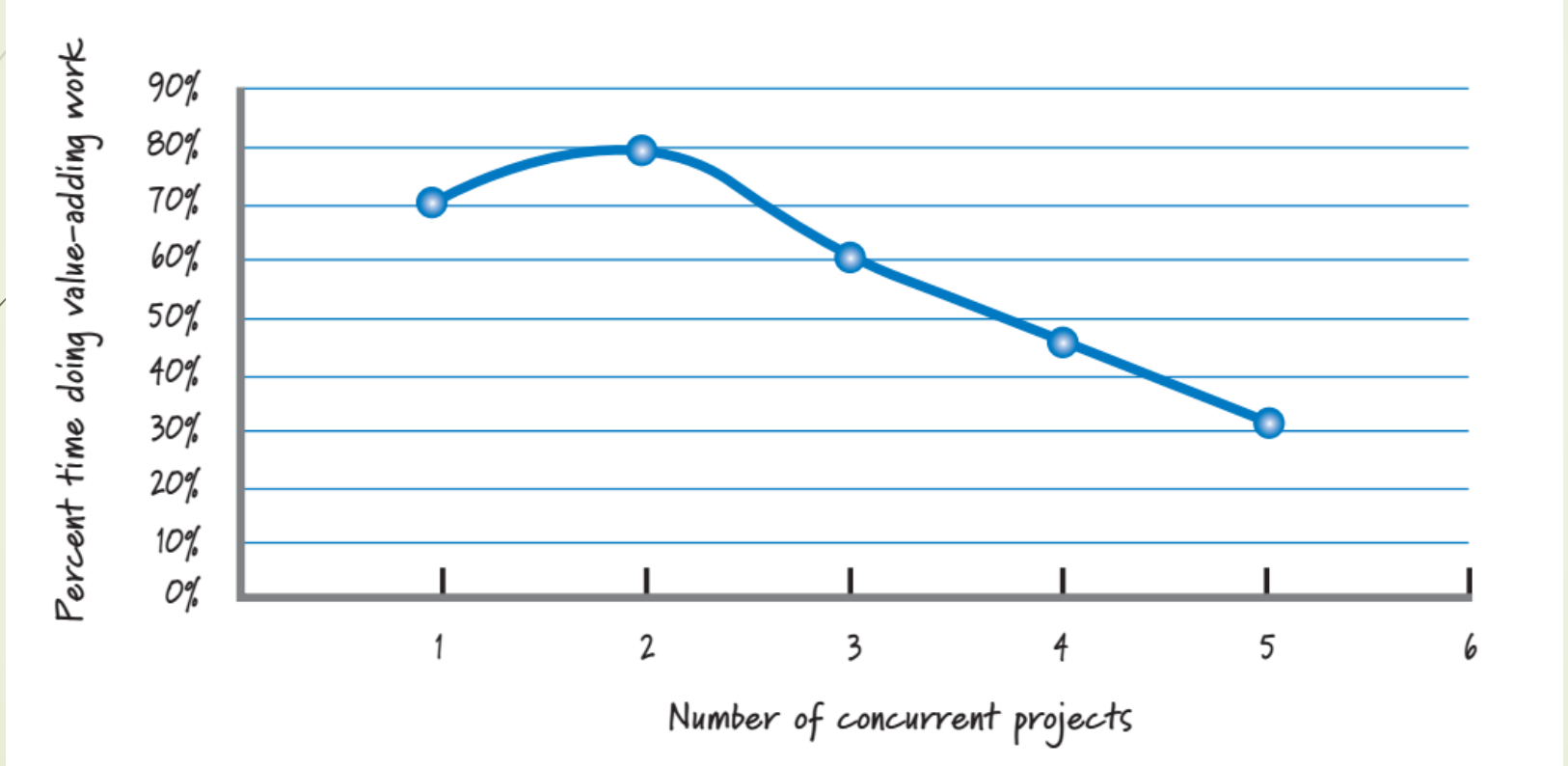
Focused and Committed

- Team members need to be focused and committed to the team's goal.
- Focused means that each team member is engaged, concentrating on and devoting her attention to the team's goal.
- Committed means that during both good times and bad, each team member is dedicated to meeting the team's collective goal.

Focused and Committed

- If a person is working on only one product, it is far easier for that person to be focused and committed.
- When asked to work on multiple concurrent product development efforts, a person must split her time across those products, reducing her focus and commitment on all products.
- It is harder for a team member to do a good-quality job when she is hopping from product to product.
- It's even harder to be truly committed to multiple products simultaneously.
- There is considerable data to support that being on multiple products (or projects) or multiple teams reduces productivity.

Multitasking



Multitasking

- This data indicates that nobody is 100% productive—there is overhead just to be a good corporate citizen.
- Productivity actually seems better with two projects than with one. This occurs because it is possible to get blocked on one project, so having a second one to switch to allows a person to be incrementally more productive.
- Based on this data, working on three or more concurrent projects is a bad economic choice because more time is spent on coordinating, remembering, and tracking down information and less time is spent doing value-adding work.

Focused and committed

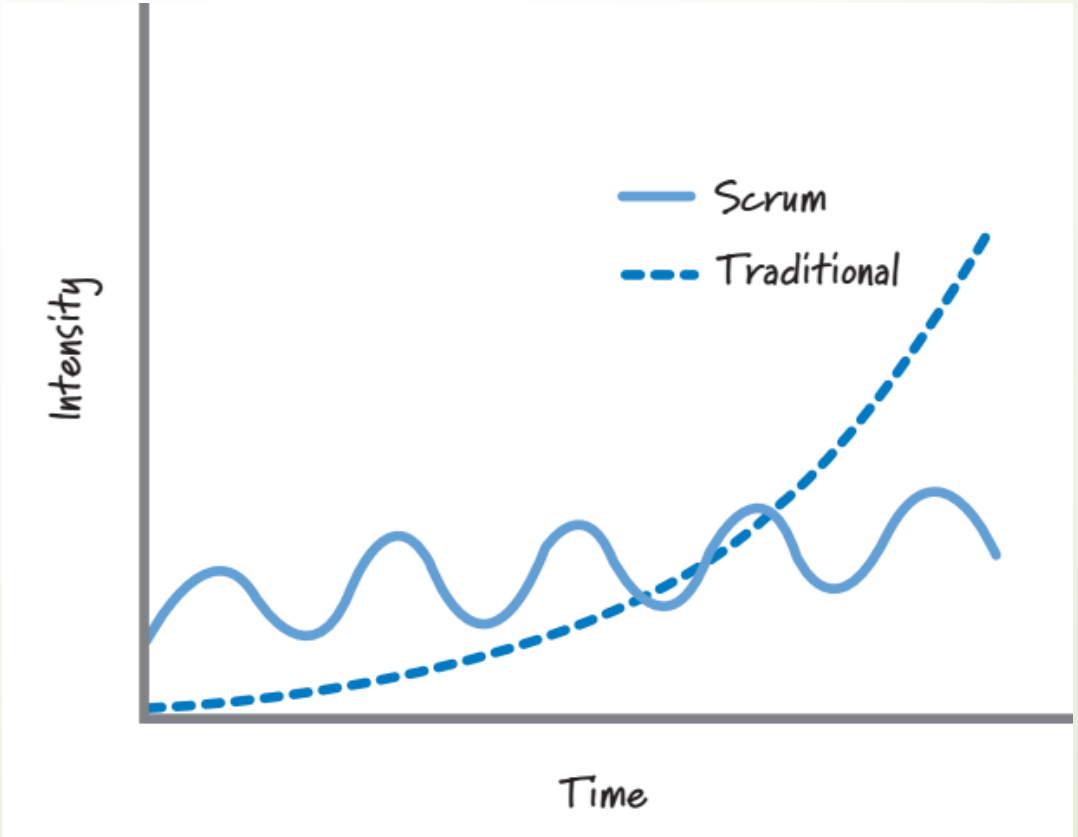
- Let the specialist decide how many products she can commit to and focus on simultaneously. If she says she can't commit to any more, don't assign her to that next product or team.

Solutions

- First, do fewer projects concurrently. This is frequently the correct solution
- Hire more specialists to share the burden.
- Help other people broaden their skill sets to include the specialty skill.
- Forcing people to work on too many projects/teams concurrently will reduce their focus and commitment and jeopardize business outcomes.

Working at a Sustainable Pace

- Team members must work at a sustainable pace.
- They deliver world-class products and maintain a healthy and fun environment.



Sustainable Pace

- This incredibly intense time is symbolized by the superheroes pulling all-nighters and working weekends trying to get out the release. Some people thrive on this type of work, love the attention, and want to be rewarded for their extraordinary effort.
- The stress on everyone else is overwhelming. As an organization we should be asking, "Why did we have to work nights and weekends, and what should we change?"
- Compare that with the typical intensity profile when using Scrum, where we've been continuously developing, testing, and integrating working features every sprint. During each sprint the team members should be using good technical practices such as refactoring, continuous integration, and automated tests to ensure that they can deliver value at frequent, regular intervals without killing themselves.

Sustainable Pace

- So, within a given sprint we'll likely see intensity increase a bit near the end of the sprint as we ensure that all work associated with our strong definition of done has been met.
- However, the overall intensity of work during each sprint should closely resemble the intensity of the previous sprint, reinforcing the team's working at a sustainable pace.
- The aggregate result is a leveling of the work; it doesn't come in huge chunks or intense bursts, especially late when it is most harmful. This leveling means that Scrum teams will likely work fewer overtime hours and therefore be less likely to burn out.

Long-Lived

- Effective use of Scrum requires teams, not groups.
- A team is made up of a diverse, cross-functional collection of collaborating people who are aligned to a common vision and work together to achieve that vision.
- A group is a collection of people with a common label. Other than sharing the group name, group members don't share much else and won't effectively fulfill the responsibilities of development team.

Long-lived

- As a rule, teams should be long-lived.
- I keep my teams together as long as it is economically sensible to do so.
- Research has shown that long-lived teams are more productive than newly formed groups.
- Research demonstrates that team familiarity (team members' prior shared work experience) can positively impact the efficiency and quality of team output.
- Improved productivity, efficiency, and quality lead to improved business results.

Long-lived

- If we start out with a group of people who have never worked together, we have to spend time and money.
- Once we have a well-functioning team, we have a real business asset. Its members know how to work together, and they have earned each other's trust.
- In addition, the team has amassed important historical information, such as the team's velocity and shared estimating history.

Long-lived

- The *currency of agile* is the team. In fact, one of the core values of the Agile Manifesto is "Individuals and Interactions."
- In other words, the team is the valuable asset.
- Moving people around from team to team destroys the integrity of the team.
- But, if we have a team that really hasn't gelled the way we had hoped, or is otherwise dysfunctional, it is often less disruptive and economically more sensible to disband the team.

References

- Essential SRUM, 2012.
- Scrum guide, 2017.