# Agile Stakeholder Engagement and Team Development

Agile projects are only successful if the Agile team is cohesive, productive, and engaged. When managing projects for your organization, you must encourage active involvement to ensure your team has a clear understanding of project requirements and stakeholder expectations.

In this course, you'll learn about team leadership and the characteristics of effective stakeholder engagement when managing Agile projects. This course also introduces you to some important Agile project management tools and techniques, such as team development strategies and developmental mastery models. Finally, this course covers effective communication tools and the benefits of using facilitation techniques like Agile games to foster collaboration and cooperation in the Agile team.

This course is one of a series in the Skillsoft learning path that covers the objectives for the PMI Agile Certified Practitioner (PMI-ACP)® exam. PMI-ACP is a registered mark of the Project Management Institute, Inc.

## Table of Contents

1. Agile Stakeholder Engagement and Team Development
2. Engaging Stakeholders in Agile Projects
3. Stakeholder Role throughout the APM Model
4. Effective Decision-making in Agile Projects
5. Agile Team Characteristics
6. Scaling Agile Teams
7. Agile Team Development
8. Developmental Mastery Models
9. Communication Tools
10. Agile Games

## Agile Stakeholder Engagement and Team Development

*[Course title: Agile Stakeholder Engagement and Team Development. The presenter is Barbara Waters, PMP-ACP.]* Having a good team and quality leadership is key for the success of any project. But in agile development, it is crucial. In this course, you'll learn about the importance of stakeholder engagement and some useful tools for engaging your agile team. You'll also learn about ways to build cohesiveness within your project team and effective team and communication development methods.

## Engaging Stakeholders in Agile Projects

*[Topic title: Engaging Stakeholders in Agile Projects.]* One of the key characteristics of an Agile community is that it relies on our cross-functional teams. This enables members to draw on one another's strengths and support throughout the life of a project.

An Agile project community is made up of the development team and stakeholders. Stakeholders are both internal and external, and are individuals with a vested interest in the success of a project, and who are affected by or have input on the project. For example, some stakeholders provide technical know-how, while others make decisions about which product features are necessary for the final product.

Internal stakeholders work for the organization, whereas external stakeholders are usually clients or customers from outside the organization. Internal stakeholders would be, for example, members of the executive team who are responsible for the project. They may make decisions with regards to investment or risk.

Other internal stakeholders include developers who aren't necessarily assigned to the team for a particular project, but they may be called upon for their experience and knowledge. External stakeholders include members of the product team. This team includes the customer or product owner and may include one or more domain experts and a business analyst. The product owner represents the customer and is responsible for promoting, communicating, and updating the product vision. The product owner creates use cases or user stories, sets priorities and reviews the validity of completed work. The product owner has to be fully committed to the project, as do all other stakeholders.

Domain experts are technical subject matter experts. The development team may rely on their expertise and advice to solve problems and build a product that's technically accurate. And a business analyst acts as an intermediary between the customer and the development team.

External stakeholders may also include end user or user groups who will use the product that's developed. These stakeholders aren't concerned with how the product is developed, but rather more concerned with what the product does, and is expected to do.

There are varying levels of commitment that you may see in stakeholders during a project. They're either committed, reluctant to commit or enthusiastic only at the beginning. Some of the common reasons that these various levels exist is because stakeholders often don't understand or haven't had the benefits explained. If the benefits of stakeholder engagement aren't explained thoroughly, it can be difficult for your stakeholders to understand why they should participate in the project.

As the project progresses, stakeholders may forget the benefits because they are overwhelmed with work, or working on multiple projects at one time. And if stakeholders aren't held accountable for participating in the development process, it's likely they'll disengage or even worse, not participate by the end of the project. If criteria or processes are poorly defined, it can quickly escalate to stakeholders not being committed to the project. If user acceptance criteria and testing processes are poorly defined, the customer may become frustrated and less willing to review the next iteration's results.

Stakeholder engagement depends on good working relationships between everyone in the project community. In turn, these relationships depend on trust. If there is little trust between developers and other stakeholders, they are less likely to communicate openly.

Of course, inappropriate tools prevent and discourage stakeholder engagement. Agile teams favor simple, highly visual tools that are easy to understand and use. Let's talk about some strategies to help keep stakeholders committed.

It is essential to get support for the project from senior management who make decisions about whether and how much to invest in projects. By keeping managers informed, you can help them make the right decision about issues such as hiring and software or other materials you may need. It's also important to train key stakeholders on technologies and processes that the team will be using. This, of course, ensures that they can fully complete or participate in their roles properly and can help give them a sense of belonging.

You need to be very flexible when managing stakeholder engagement. It may not always be possible for stakeholders to meet with developers face-to-face, as this is common in an Agile methodology. But with today's technology, there should be no reason why video conferencing isn't an acceptable alternative. And if a customer decides to assign a representative, it's important to accept this individual on the project team. The representative can play an important role, provide valuable information about the customer's needs, and answer any questions the developers may have.

In summary, stakeholders for your Agile project include both internal and external stakeholders. Stakeholder commitment is essential for your project to be successful. And in order to gain and maintain stakeholder commitment, you need to have senior management support for the project, train key stakeholders on the technologies and processes they'll be using, be flexible and accept customer representatives as part of the stakeholder group.

# Stakeholder Role throughout the APM Model

*[Topic title: Stakeholder Role Throughout the APM Model.]* One of the project leader's key responsibilities is to keep stakeholders engaged. And to manage the relationship between them and the development team. And stakeholder engagement doesn't just happen at the beginning of your agile project. The five phases of the APM model are the initiation phase, the release planning phase, the development phase, the review and adapt phase, and the closing phase. Stakeholders play a critical role in the activities that take place in each and every one of these five phases of your project.

During the initiation phase, the product owner is responsible for establishing the product vision. This is a high level description of the product that the project will deliver. Its goal is to inspire stakeholders and team members. This is to ensure that everyone has a common understanding of the product.

During the release planning phase, the team plans the project work that needs to be completed in order to develop the product. In consultation with other stakeholders, the product owner develops and maintains a product backlog. Stakeholders also help create user stories during this phase.

At the beginning of each development phase, stakeholders participate in iteration planning, or, in agile terms, sprints. *[The iteration planning consists of the following steps: Plan, Design, Build, Test, Review, and Launch.]* During iteration planning, the product owner and development team reevaluate the user stories slated to be built during the coming iteration. Once an iteration starts, some stakeholders, namely the product sponsor, simply focus on monitoring the progress of the development team's work. Other stakeholders would be more extensively involved in guiding and advising the team. Stakeholders such as a product owner, the domain expert, and the customer representative act as advisors to the development team. And they should be readily available to answer any of the team's questions or resolve problems that are of other importance.

During the review and adapt phase, the product team, managers, customers, and sometimes end users and developers from other projects, take part in product demonstrations. This is to evaluate the working features produced in terms of functionality valuable to the customer and overall quality. The development team and stakeholders hold team performance evaluations after each sprint to assess performance. Identify areas of improvement, and determine if the team is sufficiently able to respond to change. Despite agile teams being self-organizing and self-disciplined, they may need to adapt or optimize their processes so they are better able to accommodate customer requirements. Aside from performance evaluations, stakeholders also participate in status review meetings. Meeting attendants would include the executive team, product owner, and project leader so they are able to track the overall status of the project.

And during the close phase, stakeholders and the development team may hold project retrospectives, or postmortem meetings, to identify and record lessons learned, which can then be passed on to other teams. Automated surveys and evaluation forms can take the place of retrospectives. Especially if the stakeholders are geographically dispersed or busy with other projects.

In summary, agile project stakeholders are engaged and active in all phases of the APM model. During the initiation phase, stakeholders, namely the product owner, establishes the product vision. During the release planning phase, stakeholders are involved in planning the project work, developing the product backlog, and creating user stories. In the development phases, stakeholders participate in sprints, reevaluate user stories, and guide and advise the project team as required. During the review and adapt phase, stakeholders take part in product demonstrations, team performance evaluations, and status review meetings. And finally, in the close phase, stakeholders actively participate in retrospectives or post-mortem meetings.

# Effective Decision-making in Agile Projects

*[Topic title: Effective Decision-making in Agile Projects.]* In traditional project management, most decisions are made during initial project planning. But in an Agile project, most decisions are made as a project progresses and as the product develops. For this very reason, Agile projects depend on fast and effective decisions. An Agile project can't afford slow decision-making because delays in decisions mean delays in a project. Over analyzing decisions or revisiting them too many times could lead to poor decisions, which in turn has a negative effect for the project. In an Agile project, effective decision-making relies on effective participation and agreement on shared values.

Effective participation means that everyone is involved in the decision-making process, with each person having the chance to voice his or her opinion. Instead of simply delegating decision-making authority to a few individuals, participation actively seeks input from all the relevant people, and then uses this input to make the final decision.

Participation ensures that the focus is not on who makes the decision, but rather, on having the right people involved in making the decision. If decisions are to be successfully implemented, team members need to share certain values and principles. Failure to do so can result in individuals pulling the team in opposite directions, which can compromise product development and cost conflict. In an Agile project, all team members should agree on applying the Agile principles and values relevant to the team's chosen methodology.

Decision framing is a decision-making practice that places the decisions in context and assesses the overall objective. It considers who and what the decision will affect and what criteria will be used to reach the best decision. Decision framing manages the biases and emotions that individuals may bring to the decision-making process. It also stops decision-makers from rushing into making decisions. Once you framed a decision properly, you need to determine if everyone you've identified as relevant participants takes part in the decision-making process.

In order to make this decision process work, everyone's opinions first have to be heard and then discussed. This helps ensure that diverse points of view based on different areas of expertise and backgrounds are taken into account. This will result in a much better final decision. If a decision is to be accepted and properly implemented, everyone involved has to support the decision, even if they don't fully agree. As you know, this at times can be challenging. This doesn't mean you need unanimous agreement, but instead, you just need consensus. Of course, you want to ensure that this doesn't slow the process down or take up too much time. And finally, decision framing helps to build trust throughout and during the process.

A decision gradient developed by Jim Highsmith is a useful tool for Agile decision-making, as it enables members of the group to express their level of support along a continuum. *[A sample decision gradient displays.]* Keep in mind that gradients can be customized based on the preferences of the person in charge, and that ultimately the final decision is up to the decision maker. The decision gradient provides a visual representation of everyone's opinion. This tool is used to solicit everyone's thoughts and to help groups reach consensus. This involves using the traditional method of voting. Team members plot their vote or position on the gradient for further discussion.

A decision gradient is typically marked at intervals from left to right, with varying options to represent the varying levels of agreement. In Highsmith's example, there are five options. In favor, ok but with reservations, mixed feelings, disagree but will commit, and veto. Each participant places a check mark close to the option representing his or her opinion. Interestingly, when you visually plot everyone's position, you may begin to reveal particular tendencies.

For example, the majority of individuals may disagree slightly, but may be willing to accept the decision. Once results are visible, group discussion can take place. This discussion will have individuals examine why they voted the way they did and what can be done to resolve any underlying issues. The great thing about these types

of discussions is that they provide team members with a deep understanding of the issues. Which may lead them to change their minds about their initial votes.

In summary, Agile projects require fast and efficient decision-making among stakeholders. Successful decision-making technique should involve all relevant stakeholders in the process, allow all opinions to be heard and discussed. And it should result in all stakeholders supporting the final decision, even if they're not fully in agreement. A decision gradient is an effective decision-making tool for Agile projects.

# Agile Team Characteristics

*[Topic title: Agile Team Characteristics.]* Although the success of any project depends on an effective team, the role of teams is especially critical in an agile project. The reason is that in an agile environment, the team drives and shapes the project as the work proceeds. All team members take collective responsibility for a project and the associated decision-making. They also decide and agree on the standards that all members must meet.

A cross-functional team for an agile project typically involves members who have the specific and necessary skill set and discipline required to develop the product features. One of the perks to this is that there are typically less delays with a cross-functional team, because there are fewer handovers of work required. To ensure effectiveness of the team, each member's role should be clearly defined. There should be a strong leadership presence and team members should be able to consult with one another regularly.

Because cross-functional teams in an agile environment are self-organizing, the individuals performing the work will take ownership of it. Although each member manages their own work, they do share responsibility for making decisions, solving problems, and continuously improving their work processes.

One of the expectations of an agile self-organized team is to be empowered to make their own decisions. After all, who better to make decisions about one's work than the people who are actually completing it? The project leader should set the stage of empowerment, whereby the team members are free to make and learn from their own mistakes, but yet, still held accountable. The project leader should also encourage open and frequent communication. Should the team experience roadblocks or have relationship issues, the project leader must step in to assist.

Fundamental to agile development is good communication. Good communication happens when there is a level of trust and comfort amongst the team. This happens when everyone feels 100% comfortable with being able to express his or her opinions without being criticized.

A byproduct of a strong, communicative, self-organizing team is that of collaboration and cooperation. It is important that there is an appropriate balance between the two. Cooperation is based on the idea that the whole is the sum of its parts. This happens when people align their separate efforts to work toward a common goal. Collaboration is based on the idea that the whole is greater than the sum of its parts, whereby team members interact directly with one another and build on each other's ideas. This provides a rich environment where innovation can be cultivated and encouraged.

It is essential to explore an individual's personal attributes, as these can certainly have both positive and negative impact on the team and the project. Many of you have likely heard the term emotional intelligence. What does it really mean in this context? We are talking about a person being aware of and being able to control their emotions, while at the same time being cognizant of other's emotions. Someone who displays integrity is seen as dependable, accountable, honest, and trustworthy. As you can imagine, we would want a team full of people with this type of characteristic.

Team members must have a personal sense of responsibility if they are to cooperate and collaborate with others. Team members should be punctual, arriving at work with a clear mind, so that they can focus on the task at hand. People who exhibit self-confidence are genuinely more comfortable asserting their opinions, and they are often better equipped to make decisions on how work should proceed.

And finally, having team members who are readily comfortable with asking for help when required will assist with the speed of resolution.

In summary, in an effective agile team, rules are clearly defined. There is a strong leadership presence, and consulting among the team is supported. Individuals take ownership of their work. Members are empowered to make their own decisions. Open and frequent communication is encouraged, and there is a good balance between collaboration and cooperation.

# Scaling Agile Teams

*[Topic title: Scaling Agile Teams.]* Maybe you find yourself managing a project that requires a significant number of developers. Or a project where there are many teams involved, perhaps even in different locations. Not only will projects like this require extra diligence in managing the project work. But you need an effective strategy for ensuring the people working on the project, including all stakeholders, feel engaged. This can be challenging when managing agile projects with significant project team and stakeholder numbers.

Scaling agile means going from a few agile teams to multiple, or even hundreds, of agile development teams. There are some unique challenges that come up whenever you have significant numbers of people needed to work together in a coordinated fashion. Scaling projects seems to be contradictory to agile methodologies. Because the premise of most agile projects is that teams are small and co-located. However, agile teams can in fact be scaled.

Obviously one main factor to consider when scaling agile is the size of the team. But there are other factors to consider. One is the geographic distribution of the team. Maybe teams are in other cities or other countries. This presents implications when it comes to face-to-face meetings and effective daily stand-ups.

Another consideration is to ensure that your plan, during the release planning, is captured electronically so that everyone can see where it is.

Organizational distribution is something to consider. It is generally considered even if you didn't have to scale a project. Because you don't know which divisions initially would be part of the project team. But scaling just adds a level of complexity if different divisions or stakeholders are then geographically dispersed as well. And, when we look at compliance, regulatory requirements must also be very well thought out. You want to ensure that the process is within the regulation, or the end product, which adds value, meets, of course, the correct regulation.

The customer's organization, and an organization running an agile project, may both be obliged to abide by various governmental and industrial regulations and guidelines. When we look at domain complexity, there are two domains, the developer and the customer. If you're scaling agile, it can be assumed that the level of complexity of the product would then also be more complex. This would require greater exploration and experimentation to possibly having to create more prototypes and models, and possibly simulations.

And of course, technical complexity is the final thing that needs to be considered. Generally, one would think that the bigger the project, the more complex it would be. This is often the case, and as such careful consideration should be given to how the technical landscape looks. Like now and where it needs to be in order to ensure that this project can be scaled.

We also want to be very cognizant of cultural diversity, especially if the teams are geographically distributed. Culture is very important when it comes to trust, open communication, and the sharing of documentation. There is also the challenge of a product owner. One thing to consider depending on the size of the scale is that you may want to create a product owner team. With the intention of having one product owner setting priorities and mediating between stakeholders. And the other product owner for each team to help communicate what is required for each product backlog item. Again, this just might be some consideration based on the sheer size and scope of the project.

In summary, though it presents unique challenges, scaling agile teams is possible. Considerations that you should make when scaling agile include the team size, the geographic distribution, electronic documentation, the organizational distribution, compliance, domain and technical complexity, cultural diversity, and the product owner role.

# Agile Team Development

*[Topic title: Agile Team Development.]* Despite Agile teams being self-organizing and self-disciplined, leadership is still required. An Agile coach focuses on developing and maximizing both team and individual performance, and they help Agile teams apply Agile practices to their daily work. The goal of the coach is to develop both the team and individual skills and abilities, thus encouraging them to take initiative, allowing them to fail and recover and to ensure that any obstacle has been removed so that it doesn't impede their progress.

A coach is there to help facilitate the working relationship with the team members, whether this includes helping with communication skills or uncovering any roadblocks. Coaches should facilitate relationship building between the development team and the customer. And sometimes when people, not only Agile team members, get working on the tasks at hand they sometimes lose sight of the goals of the project. The role of the coach is to ensure that this doesn't happen.

It's important for project leaders to recognize when a team isn't performing up to snuff. It's then up to the leader to implement the necessary strategies to boost team performance. One of the ways to do this is to set high performance and technical expectations as a benchmark, simply refusing to accept anything but the best from the team. Most people will generally rise to the occasion of these expectations. The leader may need to provide a little extra nudge to some to ensure that they are functioning at optimum capacity. This involves removing any obstacles that could hinder the delivery of the finished and useful product by the end of each iteration.

Another strategy includes recognizing individual efforts and contributions. Individuals working in Agile teams need to know that the work that they are doing is meaningful and worthwhile. This is because most teams are driven by a quest for knowledge and success. Providing a little recognition during product demonstration can certainly go a long way to making them feel recognized, and providing a valuable contribution.

Another way to improve in performance is to quickly identify problems so that they can then equally as quickly be resolved. This is advantageous to improving the team performance as it removes and reduces waste.

Ensuring team members are energized is yet another useful strategy for improving team performance. To remain energized, team members need to maintain the proper work life balance. Occasionally, team members may be required to work overtime to meet a deadline. But a need for regular overtime indicates something different, such as poor time management, and this should be monitored and evaluated. Otherwise, extended overtime could lead to burnout, increased staff turnover and a reduced quality of work.

And finally focusing on one task at a time is vital. Multitasking, or working on more than one project or task at a time, is often thought of as a way of maximizing the efficiency. Well, that's not the case in Agile. It is accepted that multitasking actually reduces productivity. This is because it takes time and effort to shift between subject matter and different technologies. Using an Agile approach, a team focuses on a small set of tasks at one time.

In summary, there are several strategies you can use to improve team performance on Agile projects. These include setting high performance and technical expectation benchmarks, recognizing individual efforts and contributions, quickly identifying problems, ensuring team members are energized, and having team members focus on one task at a time.

# Developmental Mastery Models

*[Topic title: Developmental Mastery Models.]* Developmental mastery models are useful approaches for individual and team development in your Agile environment. While there are many different models out there, three common developmental mastery models particularly effective in Agile projects are the Shu-Ha-Ri developmental mastery model, the Dreyfus model of skill acquisition, and Tuckman's model of group development.

The Shu-Ha-Ri developmental mastery model comes from Aikido, which is a form of Japanese martial arts. This model has been adopted as a way of thinking about learning techniques in an Agile environment. The concept is that a person passes through three stages of gaining knowledge.

The Shu, the beginning stage, is where the learner follows or copies the teachings of one's master precisely. The learner concentrates on how to do a specific task without worrying about how much the underlying theory applies. If there are multiple variations on how to do a task, he or she will simply concentrate on just one way, the way that the master taught it.

The Ha, the second stage, is where the learner starts to branch out, and try variations of the technique. So in an Agile environment, for example, the Agile team will be looking towards improvement during retrospectives. Once a process is followed, adapting can be accommodated to know the best fit for an organization. It's logical to have a deep understanding of a process, and understand what is working and what isn't before trying to branch out and trying variations. In the Ha stage, organizations try to figure out what works for others, and then learn from it.

And finally, in the Ri stage, the learner isn't learning from other people, but instead, the learner is learning from his or her own practices. They create their own approach and adapt that to what they have learned to their particular circumstance. Here, the organization evaluates their own experiences, and learns from them.

The Dreyfus Model is a helpful way of looking at how people acquire and grow their skill sets. *[A sample Dreyfus Model displays. The model is represented in the form of a pyramid, which is divided into five parts, each part representing a stage of the Dreyfus Model. The five stages of the Dreyfus Model are: Novice, Advanced Beginner, Competent, Proficient, and Expert. The Expert stage is at the apex and the Novice stage is at the base of the pyramid.]* Understanding how people learn is important for Agile teams, both those starting to adopt practices, and those who are already applying them. This model can be used in Agile as an evaluation and coaching tool. Dreyfus said that to become competent, you must feel bad. But what does this actually mean? Well, as one grows their skill set, they grow through the various stages laid out by Dreyfus. And as you begin to become more experienced, to the point of becoming competent, you start to feel bad or become overwhelmed, due to the number of potentially relevant elements or pieces of a project that you are now able to see. To advance and move through the stages of learning a new skill, you'll require clear instructions, rules, and boundaries. You will then begin to practice on your quest to attain new skills.

As a learner moves from abstract concepts to concrete experiences, they will advance more readily through the stages. It is emphasized that concrete experience is where learning begins. The model demonstrates how a person moves through the stages on a scale, by starting to use a practice *[The Novice stage and the Advanced Beginner stage of the Dreyfus Model are highlighted.]* and using a practice effectively *[The Competent stage and the Proficient stage of the Dreyfus Model are highlighted.]* and to the point where they are effectively applying the practice without conscious effort. *[The Expert stage of the Dreyfus Model is highlighted.]*

The Dreyfus model is based on four binary qualities. Recollection, which is either non-situation or situational. Recognition, which is either decomposed or holistic. Decision, which is analytical or intuitive. And awareness, which is monitoring or absorbed.

Forming a team takes time, and members often go through recognizable stages as they change from being collections of individual strangers to a cohesive team with a common goal. Bruce Tuckman's forming, storming, norming, performing, and adjourning model describes the various stages that individuals who gather together go through to form teams.

In the forming stage, team members are brought together. You have different levels of enthusiasm, just as there are different personalities and skills that make up the team. This is an important stage where the project leader needs to ensure that there is a clear definition of each person's roles or responsibilities within the sprint.

In the storming phase, people start to push against the boundaries established during the forming stage. This is a stage where there is a potential for the team to fail. Team members have formed opinions about their team mates, and generally, people's personal goals are taking precedence over the team goals. As a project leader you'll want to coach team members in assertiveness and conflict resolution training and show them and demonstrate the appropriate skills. By the way, the storming phase is a normal stage in team development as the team progresses. The objective is to work through it effectively.

Gradually, the team moves into the norming stage. This is when people start to resolve their differences, appreciate colleague's strengths and respect the team. They are focused on how to apply their strengths to the benefit of the team, and help achieve the goal of the current iteration.

The team reaches the performing stage when hard work leads, without friction, to the achievement of the team's goal. The structures and processes that the team has set up support this as well.

In the adjourning stage when project work is complete, the project team disassembles and everyone goes their separate ways.

In summary, there are three common developmental mastery models in Agile. Shu-Ha-Ri developmental mastery model, which outlines three stages of learning, imitate, understand, self-directed innovation. The Dreyfus model of skill acquisition, which focuses on four binary qualities, recollection, recognition, decision, awareness. And Tuckman's model of group development, which outlines the five stages of team development, forming, storming, norming, performing, and adjourning.

# Communication Tools

*[Topic title: Communication Tools.]* In agile methodologies, knowledge sharing occurs largely when individuals interact with each other face-to-face. Agile methodologies also promote knowledge sharing as an essential part of the project's success. Successful knowledge sharing depends on successful communication. Particularly, the free flow of information. There are several tools to encourage the free flow of information.

The first is flowcharts. Flowcharts are typically used to describe the detailed logic of a business process or a business rule. The best way to utilize flowcharts is to keep things simple. Work together with your stakeholders to sketch the flow of the business process on a whiteboard and discuss it. If you like, you can then take a digital photo and save it so that you can refer to it later. The value often isn't in the model itself specifically, but instead in the whole act of modeling because it helps you think things through.

User story cards are another effective communication tool. User stories are typically done on index cards and are used to explore how people will use the system. They are used at various times throughout the project phases. They are typically prioritized on a scale of one to ten. And then they are moved around the stack to demonstrate the priority level. The card also shows a unique ID. This is for traceability between the user story and other artifacts later on.

Burndown charts show project status at a glance. Time is shown on a horizontal axis, and the number of features still yet to be completed are displayed along the vertical axis. *[A sample Burndown Chart titled Product Release Burndown displays. Week is shown on the horizontal axis and Story Points is shown on the vertical axis.]* The actual chart, again, should be placed in a location where everyone on the team, internal and external stakeholders, are able to visually see where the project is and what is yet to be completed.

A task board can be used to visually represent all the work that is being done by the team. *[A sample Task Board displays. It shows the following four statuses of work: To do, In progress, To verify, and Completed.]* So they are

more complex and versatile. A physical task board is really a living entity that has to be manually maintained. Otherwise, it doesn't serve the purpose of being useful.

Planning boards, as the name would suggest, make it exceptionally simple to see what is going on in a project through a quick glance.

And finally, storyboards. *[A sample Storyboard displays. It shows the following three statuses of work: To do, In progress, and Done.]* They are similar to just a scenario. They illustrate the interaction required to achieve a goal. But instead of using a list of steps, a storyboard actually visualizes the interaction. A storyboard can help a developer quickly get a sense of what work still needs to be completed.

Geographically dispersed teams require additional communication tools. When agile team members are geographically separated, the need for documentation may be greater than ever. Keep in mind, though, that documentation should be used sparingly as it can sometimes complicate the communication.

Documentation information is often misunderstood, because understanding often depends on a combination of content and context. And text alone is sometimes a poor conveyor of context.

Finally, when working with dispersed teams, you may need to bring in social media tools. Or other online tools that will make it much more efficient and useful for the teams to collaborate and work together. The whole point is to ensure that there is open and free-flowing information.

In summary, there are several communication tools you can use for information flow in your agile team. Including flow charts, user story cards, burndown charts, task boards, planning boards, storyboards. Written documentation when necessary, and social media and online collaboration tools.

# Agile Games

*[Topic title: Agile Games.]* Facilitation techniques, otherwise known as games, have a variety of uses in an Agile environment. While at first they may sound childish, games can be used to model complex, time-consuming processes in a fun and light-hearted way. Or facilitate communications amongst team members who don't typically communicate with each other, or about the same things.

Agile games have several benefits. They can be used to foster collaboration, communication, and innovation in Agile teams. And Agile games can be used to teach, demonstrate, and improve teams overall.

Using games helps to drive good behaviors, including cooperation, clarifying, inspiring, risk taking, harmonizing, and process checking. All the while helping to overcome the destructive team behaviors of dominating, rushing with drive, or digressing, discounting, or blocking.

There are several types of games. These include collaboration activities, brainstorming activities, such as affinity mapping, contrasting variants games, where two variants are compared, and retrospectives activities.

There is a game called The Learning Matrix which is used to find things to improve upon based on looking at what it is that you're trying to do now. Other game activities include drawing and storytelling.

In summary, Agile games are effective facilitation techniques that provide many benefits. They foster collaboration, communication, and innovation. They can be used to teach, demonstrate, and improve processes. They also help model complex processes, facilitate issue examination, and improvement identification, drive good behaviors, and help overcome destructive behaviors.