

# AGILE GUIDE

## Table of Contents

<i>Introduction</i> .....	3
<i>Characters</i> .....	4
<i>Definitions</i> .....	5
<i>Meetings</i> .....	6
<i>Chapter 1</i> .....	8
<i>Servant Leadership</i> .....	8
<i>Chapter 2</i> .....	10
<i>What the Business Wants from You—Managing Requirements</i> .....	10
<i>Chapter 3</i> .....	12
<i>Your Agile Team</i> .....	12
<i>Chapter 4</i> .....	14
<i>The High-Performance Team</i> .....	14
<i>Chapter 5</i> .....	16
<i>Everyone around the Campfire</i> .....	16
<i>Chapter 6</i> .....	21
<i>Daily Stand-Up, or Daily Scrum</i> .....	21
<i>Chapter 7</i> .....	24
<i>Introducing the Product Owner, or Value Driver</i> .....	24
<i>Chapter 8</i> .....	27
<i>Discoveries from the Product Backlog</i> .....	27
<i>Chapter 9</i> .....	31
<i>The Sprint Backlog and Release Planning</i> .....	31
<i>Chapter 10</i> .....	35
<i>Sprint Planning Meeting</i> .....	35
<i>Chapter 11</i> .....	38
<i>User Stories and Estimation</i> .....	38
<i>Timeboxed Sprints (Iterations) and the Meaning of Done</i> .....	46
<i>Chapter 13</i> .....	50
<i>Tracking Flow and Information Radiators</i> .....	50
<i>Chapter 14</i> .....	54
<i>Demonstration of the Product</i> .....	54
<i>Chapter 15</i> .....	57

<i>The Retrospective</i> .....	57
<i>Chapter 16</i> .....	60
<i>Wash, Rinse, Repeat, Win!</i> .....	60
<i>Chapter 17</i> .....	63
<i>Team and Business Cultural Dynamics—Team Science™</i> .....	63
<i>Example Case</i> .....	65
<i>Chapter 18</i> .....	66
<i>Scrum of Scrums</i> .....	66
<i>Example Case</i> .....	66

# Introduction

## Team Tribes: A Story

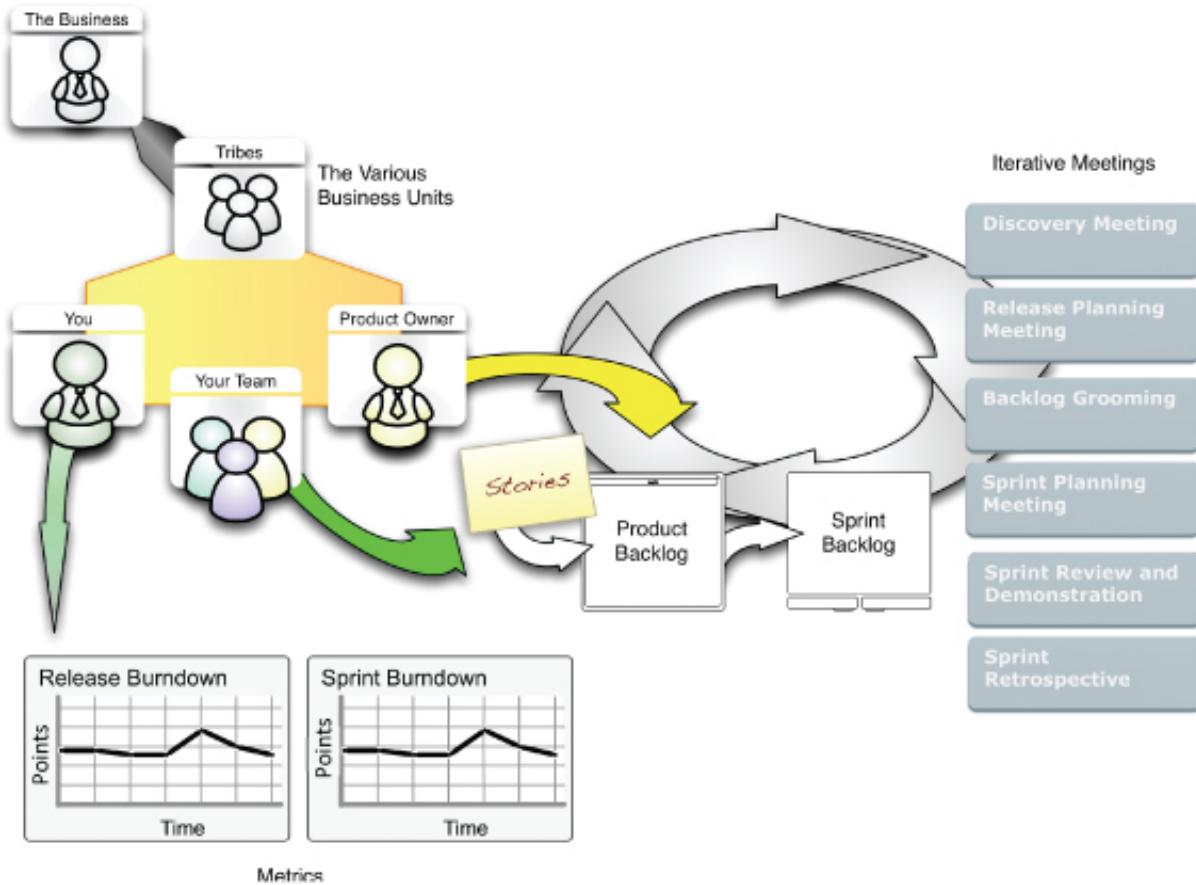
The business jungle is as vast as it can be dangerous. Many different Tribes, or business units, inhabit the jungle, and they rarely interact with one another unless they are trading goods and services, seeking to expand their influence, or fighting. The landscape of the jungle is littered with once-meaningful artifacts and orphaned documents, as well as the remains of unfinished projects and the ashes of currency and project funds burned away. An overall fear of the unknown marks the well-tread paths that lead to each Tribe. This is not a safe jungle, and if you were to go off the beaten path, you would find the warning markers of imminent danger and the tombs of those who tried to change culture. Members of some departments often visit the shrines that surround the mystical waterfall, where the cadenced flow of the water reminds them of well-instituted processes and procedures that tend to keep the world in balance. Each business Tribe is different, with its own culture and history, and each has its own fears and reservations about change.

A typical business Tribe works very well within its own boundaries and influence. Each business unit contains a leader-diplomat who goes out to work with the other leaders of the different Tribes. The leader-diplomats are usually pillars within their departments and are well equipped with the knowledge of how things are done. They are the keepers of the status quo, and they can exert great influence over others when they see an opportunity to make better use of the Tribe's talents.

You have been selected as the next Leader or ScrumMaster for one of the most reclusive and curious teams in the jungle: the Agile Team. In the recent past, this Tribe has time and time again shown the other members in the business its effectiveness in completing goals. When trading with other teams, leaders in application development see your Agile Team as the most cost-effective Tribe in the land. The leaders of other departments are becoming more aware of the Agile Team and have taken quite an interest in how they may be able to leverage or utilize the often-misunderstood yet awe-inspiring, up-and-coming Tribe. Less and less are the leaders going to the mystical waterfall for guidance; they are now coming to you for help. The time is ripe to establish your team. You are the Leader who will adhere to the principles that make your Agile Team successful. Ready? Go!

Any story about Agile, Scrum, and Team Tribes will include a unique cast of characters who define their own terms, and who engage with each other as they work through a cycle of various events (i.e., iterative meetings). (See Figure i.1.)

**Figure i.1** The Scrum Process



## Characters

**Agile Team:** Your cross-functional Agile product or software development team.

**Leader:** The project/program leader, ScrumMaster, or coach for the Agile Team.

**Product Owner:** The person responsible for defining the requirements and Product Backlog (often known as “the client”), who prioritizes and grooms user stories for readiness.

**Stakeholder(s):** Business individual(s) to whom the Product Owner reports.

**Tribes:** Various business units and stakeholders within the business.

# Definitions

Agile: A framework for delivering products quickly and efficiently.

Cross-Functional Team: A team made up of individuals who can wear many different hats and fulfill many different roles (i.e., a developer doesn't only code, he also does quality assurance and analysis).

Epic: A large requirement or composite user story that needs to be broken down into small segments during software development.

Iteration: A set period of time designated as part of the incremental process of developing and producing the software using an Agile framework (see Sprint and Timebox).

Product Backlog: A list of all user stories for a product.

Release Burndown: Measures the entire backlog of items over the course of the release plan.

Sprint: A set period of time in which to complete a task (see Iteration and Timebox).

Sprint Burndown: Measures the completion of items over the course of a sprint.

Sprint Backlog: A list of user stories for a specific sprint.

Sprint Review: A demonstration of the product at the end of a sprint.

Theme: A group of common user stories for a sprint or iteration that the team wants to deliver.

Timebox: An Agile software development tool that allows your team to manage a Sprint and determine the appropriate period of time in which to develop features and user stories.

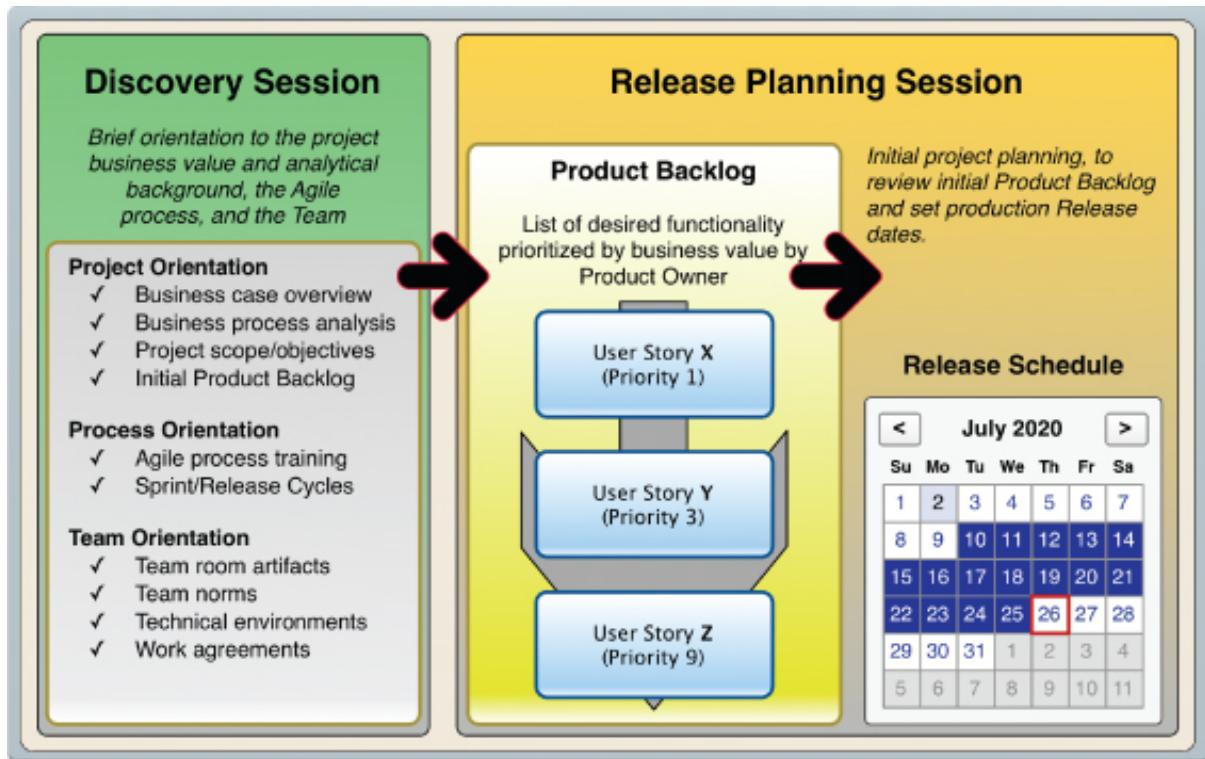
User Stories: Functional units of work (i.e., software requirements to satisfy, such as adding a pull-down menu to a website).

Velocity: A metric for measuring the rate at which teams deliver business value (add up features delivered over an iteration). Often viewed in a Sprint or Release Burndown Chart.

Waterfall: A very defined software development methodology defining *progress* as a steady downward flow that requires completion of a process before a next process can begin.

Your Agile Team will go through several discovery and planning processes before a software product is ready for release. (See Figure i.2.)

**Figure i.2** Overview of Discovery and Release Planning Process within Scrum



## Meetings

Discovery Meeting: Initial introduction to a project.

Refactoring Meeting: Review of code and work that needs to be fixed from a previous iteration.

Release Planning Meeting: High-level breakdown of project sprints and delivery dates.

Sprint Planning Meeting: Breaking all stories down into Timeboxed iterations.

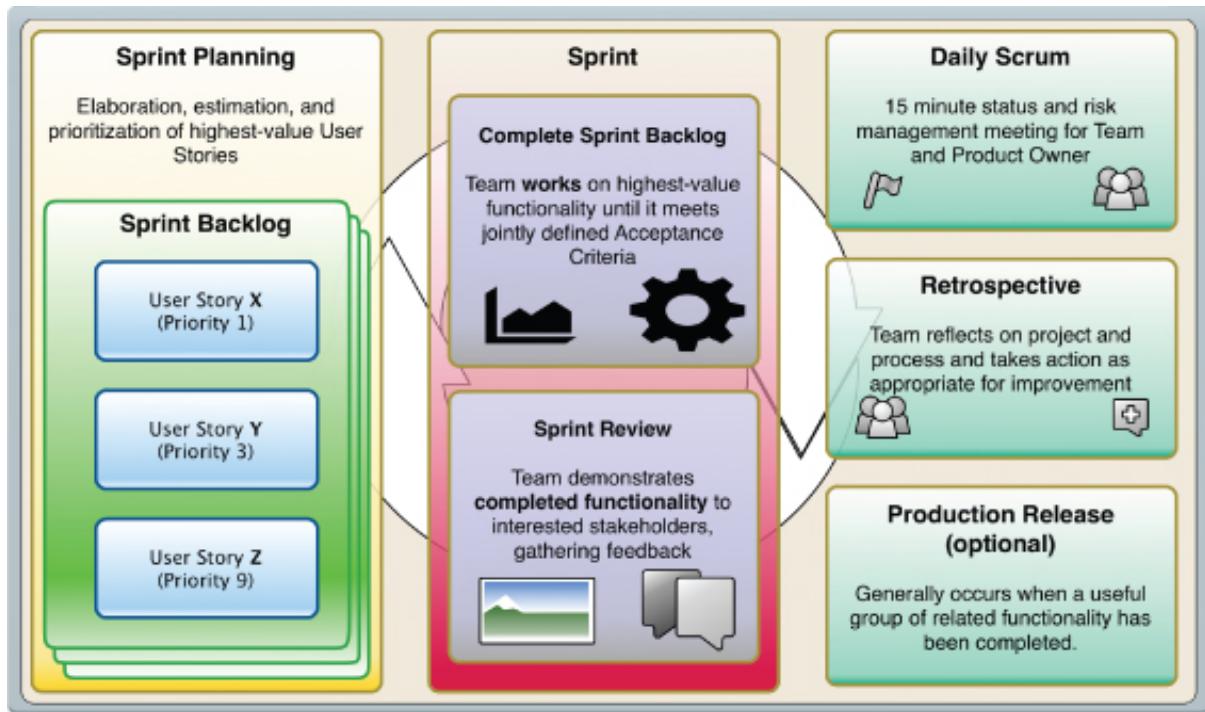
Sprint Review Meeting: A time for the team to review and show work completed during a sprint.

Sprint Retrospective Meeting: Review of a previous iteration.

Backlog Grooming Session: Functional breakdown of project items or requirements into developable or actionable units of work.

Sprint cycles, including the Daily Scrum, are crucial to the iterative framework of Agile software development. (See Figure i.3.)

**Figure i.3** Overview of the Sprint Cycle within Scrum



# Chapter 1

## Servant Leadership

You have been chosen to be the next leader of the Agile Team simply because you embody three basic attributes that any successful leader needs: servant leadership, trust-building skills, and an awesome ability to communicate and facilitate. Your ability to lead and coach others springs from your humility and accommodating attitude. You are willing not only to do what it takes to be the best but also to take the low road and serve others through your work. People know they can count on you, and your ability to hold and regain people's trust enables you to build a team of trust, transparency, and accountability. You are a great communicator and protector of the team. Your ability to resolve conflicts among team members as well as business folk allows you to command an audience that trusts your judgment and relies on your ability to protect the team's work from outside influences, filtering only that which is beneficial and helpful to the work at hand. You are an educator in that you coach and teach the business people and product owners how to supply your team with the right amount of information and feedback. You are not haughty, nor are you overbearing. You are the right person for this job.

Your job is to:

- Remove obstacles or resolve dependencies between team members and teams.
- Remind the team of mission/value statements for each project.
- Ensure that the team adheres to the defined rules of Scrum or processes accepted by the team.
- Protect the team and filter nonessential information and meetings.
- Give updates and information regarding enterprise releases or project updates.
- Set the example for the team and for the business.

A wise person once said, "Where there is no vision, the people perish." As the leader you hold the true mission and vision of the business in your hands. Your knowledge of how your team's work directly impacts the

overall business is crucial to making sure that anything your team accomplishes falls in line with your business's model for success. Your team will look to you for that direction. They will look to you for example. They will trust your decisions. They will follow your lead.

## **Leader Questions**

Every chapter in this book ends with a set of three questions you will want to ask your team.

- 1.** Who do we need to meet with or connect with to help with that task?
- 2.** Is there anything we've missed or not considered?
- 3.** What do you need from me, and what can I do to help?

# Chapter 2

## What the Business Wants from You—Managing Requirements

The jungle is perilous, fraught with danger, and always in a state of change. In your jungle, the business comprises many different tribes, teams, or departments. Each of these units within this business has its own personality and history. Even though the business is one major unit, the factions give rise to warring with one another and fighting for the same resources—namely, your Agile Team.

The jungle is not a safe place for even trade. The leaders of each tribe try to work with one another to find a common ground, and they work with you as they give rise to half-baked parchments of work (requirements). The marketing tribe to the sales tribe, the finance tribe to the customer service tribe—all need to utilize your team in order to get what they need done. Unfortunately, they don't always know what “done” is, and they often give you incomplete or nebulous requirements. The tribes have heard that your new Agile Team is taking a new approach to development that is agile, flexible, transparent, and adaptable. How will you communicate to the business that your tribe and Agile practices will knock their socks off? How can you convince them that taking an Agile approach is far better than the mystical waterfall processes and approaches?

So why are Agile requirement processes and change management good for the tribes?

1. Stakeholders and interested parties get to work together to build the right requirements.
2. Stakeholders get constant daily feedback on updates.
3. Stakeholders have control over the scope: they can change requirements and priorities, add new requirements, and modify the product backlog.
4. Stakeholders have control over the schedule (they can fund the project iterations as long as they want).

## **5. Stakeholders have control over the budget.**

The other tribes or business units are excited to hear that they can heavily participate in many aspects of the predevelopment process, but it is your job to remind them that they are responsible for making decisions and providing information and requirements to your team in a timely manner. From here, your team will be able to communicate your time constraints, requirements, questions, and/or needs to the business in a timely manner as well. At the end of the day, the business must own the prioritization of the requirements.

## **Leader Questions**

- 1.** Do you currently have a clear understanding of the product landscape or have a product strategy map or product portfolio defined?
- 2.** Which features or projects does your business want first?
- 3.** Which features or projects will provide the most measurable business value?

# Chapter 3

## Your Agile Team

Your Agile Team is successful because of the unique individuals who make up the team. These members are the heart and soul of your success. Some things to encourage your team with are the continuing autonomy and freedom you give them to do great work. This self-organization is a principle that brings the team together in unity, in that they are generalists and specialists at the same time. The team has the right to take requirements and specifications and build pieces of the total system in the way that they see fit.

There are two ways to approach managing your Agile Team: task-oriented and relation-oriented. Your success as a team Leader hinges on the help you can give to team members as they learn to relate to each other, not just on the help you provide for tasks you assign. The amount of support you give your team and the communication methods you use are just as crucial to your team's success as improving the lack of collaboration and feedback that they probably endure in a non-Agile environment. One of the most important but often forgotten steps is the need to actively listen to your team. As a Leader, you should plan to learn from your team, and one of the best ways to learn is to listen to what they have to say. The best teams are small in size, five to nine cross-functional members work together like a charm!

Teams organize work and tasks among themselves using the following qualities and/or skills:

- Accountability: Team members are held accountable for their work and are reviewed by each team member.
- Teamwork: Team members can work together to utilize their own talents in any project or activity.
- Adaptability: Team members are able to make better decisions based on their own skills and are able to adapt to changes more quickly.
- Collaboration: Team members can circulate knowledge and experience more expediently and make decisions together.

- Communication: Team members who work closely together can communicate and collaborate on work more efficiently.

This culture of empowerment enables team members to look at their responsibilities as more than just a defined role. Each team member can draw from personal experiences, motivations, and passions. With this empowerment comes the ability for each member to have control over how he or she works and to have a purpose of aligning individual aspirations not only with the company goals but also with personal goals and personal growth strategies. One way to elicit this type of information is to sit down individually with each member and ask a few questions. These questions require you to go beyond the call of duty. It takes time and dedication to know your team and determine how individual skills or passions could be matched with business opportunities. To get to the heart of your team culture, you must know your team. (See Chapter 17 for more information on how to understand your team's culture quickly and efficiently.)

## **Agile Team Questions**

1. What are your roles and responsibilities, and how comfortable are you in them?
2. What works and what does not work with your team?
3. What are other areas that you are interested in helping with or learning about?

# Chapter 4

## The High-Performance Team

Having a supportive and caring environment to work in may sound simplistic, but whether you're in preschool or a Fortune 500 company, it makes sense. One thing you continually need to communicate to your team is that you and your management care about the relationships within the team. Not only do you care, but you also have high expectations for the team given that the correct structure and guidance are in place. This is also an environment that gives opportunities for meaningful participation in any project and gives recognition to those who do well.

In this context one word comes to mind: *differentiation*. Some people may cringe at the sound of it, but the process it embodies is exactly what will move a team from underperforming (or keeping the status quo) to doing great work and excelling at any project that comes its way. As a proponent of differentiation, I allow the team members to establish themselves in roles that uniquely fit their strengths. (See Chapter 17 for more information on how to build and assemble a high-performance team.) This takes a ton of work from you: you must recognize that there are men and women who keep the lights on and there are those who need to be moved or released back into the jungle. You must be able to spend the time necessary to tap into each team member's potential. People want to flourish and they want to succeed, and to accomplish that, a leader must separate the wheat from the chaff and retain the best talent within a team. Team members will also appreciate your willingness to help them become more productive and valuable to the business, because if you foster an environment of community, collaboration, and caring, you enable members to give feedback on what is working and what is not working, as well as on who is working out and who needs to move on.

High-performance teams are a community. A community environment:

- Breeds group interdependence, which in turn increases the success of individuals as opposed to relying on authoritarian control (top-down management or command-and-control management).
- Enables the team to set goals and solve problems together.
- Continually monitors and assesses work progress.
- Celebrates achievements and rewards individuals.
- Decreases managing overhead by the Leader, thereby enabling the Leader to focus on road-mapping work and giving specialized help as needed.
- Encourages training members as a group, which promotes a higher work ethic and increased productivity.
- Produces measurably great results over time.
- Differentiates roles that people are uniquely fit to fill (see Chapter 17 for more information).
- Coaches the team to give candid feedback and support to other team members.
- Keeps the best talent, moving or removing unproductive members of the team.

A high-performance team is one that embraces the Toyota way of *kaizen*, which means “continuous improvement.” Your Agile Team will be one that has members who continually communicate with one another, continually improve themselves, the team, the processes, and the business. Most of all, your team, through its success and transparency, will honor its commitment to the business stakeholders and grow trust between Tribes. Your Agile Team may be ready to move in this direction; if this is the case, task the members to consider their readiness and willingness.

## **Leader Questions**

- 1.** Do we have the processes and tools in place to build this type of community?
- 2.** Are we physically located in an area that promotes communication and collaboration?
- 3.** Are we meeting regularly enough to go over existing problems, improve existing processes, and be aware of current and future needs?

# Chapter 5

## Everyone around the Campfire

Communication is key in the Agile Team. You will want to remove any barriers among team members so that everyone has open lines of communication. The word *campfire* is often used to suggest an open-office environment in which the team can operate so that communication and collaboration are at its most effective levels. This kind of work environment is very unlike that of a cube farm, especially because it does not encourage isolating its members.

You may have to do some heavy lifting to move your team to an open campfire atmosphere. You may find resistance from some very territorial members, and you will very likely find resistance from any who have been around for a long time. We are creatures of habit; but in this case, everyone must be involved, and you must foster the ownership of this initiative with your team. Begin the conversation with your team about how this may happen. Team members may have some great ideas about how to collocate as many parties as possible in one centralized area. Some may not want to move, and I've had more than my fair share of experiences with these types of individuals. Don't worry! They will find out that they've become major blocks to progress, or else they'll find themselves moving on to different positions or companies. W. Edwards Deming, a U.S. statistician and author who taught top management how to improve the quality of products said, that "learning is not compulsory, but neither is survival." These words resonate with most people, and in due time the pragmatic value of moving closer together will be apparent for your entire team. The benefits of an open campfire or open office are as follows:

- All team members are in close proximity to one another; the closer, the better. Facing one another helps, too. (See [Figure 5.1](#).)
- All team members are in one centralized area where most of the work is done.
- All team members have access to whiteboards or walls. These walls are used to display charts (information radiators), whiteboard

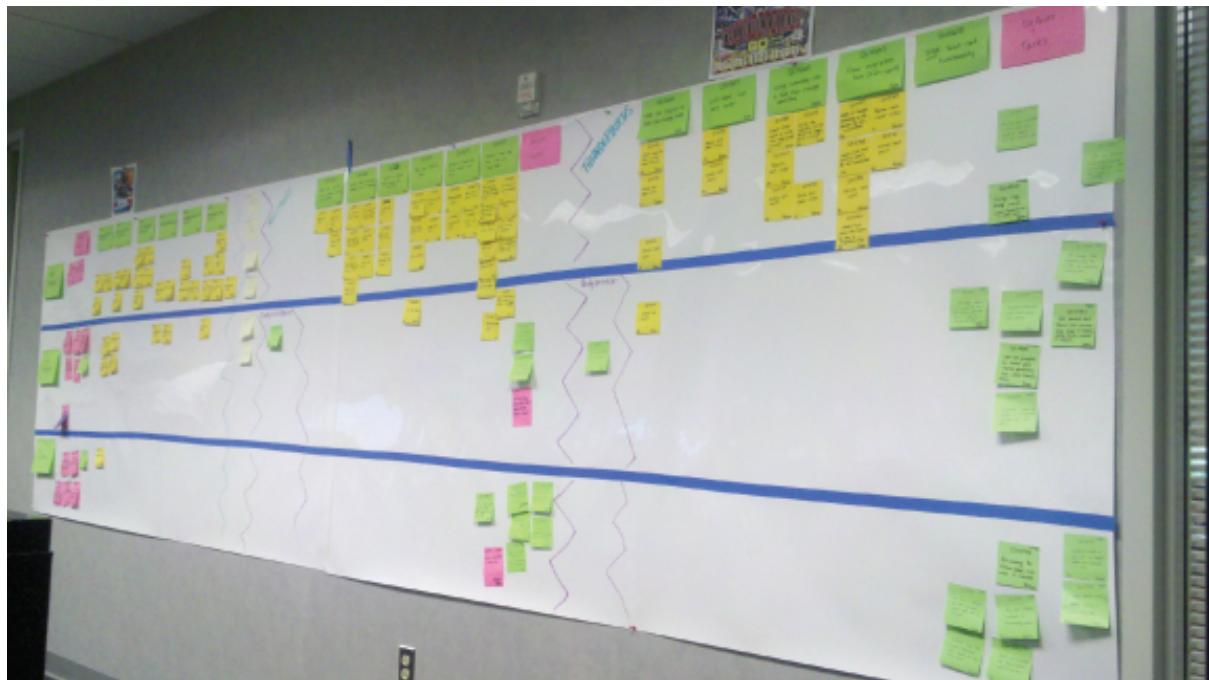
ideas, or brainstorm ideas with sticky notes. (See [Figures 5.2](#) and [5.3](#).)

- All team members have access to individual privacy areas for “heads-down” work or personal time.
- All team members are encouraged to work with their team to build an optimal work environment that makes sense for them.

**Figure 5.1** Collaborative Common Areas for Team Members



**Figure 5.2** Big Wallboards for Transparency and Team Alignment



**Figure 5.3** Information Radiator with an Electronic Task Board



Sometimes we have augment or dispersed team members that are not local to the main Agile Team. Offshoring comes to mind as an example. Distributed teams, or teams that are not physically located together, can work together with collocated (local) teams as long as they adhere to the community principles of the high-performance team discussed in

Chapter 4. Take your time to figure out which practices work and which do not. If each team member has a mind-set of continual improvement and constant communication, then the Agile Team can eventually determine the most effective practices within your development environment.

There are also communication software packages available for purchase that allow dispersed teams to communicate with one another. Again, take your time figuring out what works best within your organization and team. Remember, software platforms do not replace methodologies and good practices and/or processes. Software shouldn't be the first go-to response for any issue before fully examining the methods and processes in place. The questions for your team are meant to build up unity and cohesiveness toward this open-office goal. The more support you have from the people doing the hard labor, the more this activity of moving around desks or breaking down cubicle walls will be seen as a team project.

Go to [www.agilescout.com/best-agile-tools](http://www.agilescout.com/best-agile-tools) to find a list of more than 100 collaborative and Agile and Scrum tools to use for your team.

## **Agile Team Questions**

- 1.** What are the potential benefits and drawbacks of having an open office?
- 2.** What would be the most effective way of moving ourselves to an open-office environment?
- 3.** What are the roadblocks that must be discussed and overcome to enable us to have an open office?

# Chapter 6

## Daily Stand-Up, or Daily Scrum

Your Agile Team is more effective than any other Tribe out in the jungle because your team members constantly communicate with one another. One of the best ways to encourage this is through a Daily Stand-Up, or Daily Scrum. This is a daily morning meeting that facilitates team communication, fosters teamwork, discloses details of the project work, and highlights items for review, action, or execution. Every member of the project is invited, and every member of the Agile Team must participate; those who are outside of the core team should be silent participants. This stand-up meeting fosters shared accountability; it allows all members to be in tune with the challenges of the day and respond quickly to those challenges. This meeting is conducted standing up for a reason: it shouldn't last for more than 15 to 20 minutes.

(See [Figure 6.1](#).) Sometimes the team will go into more detailed discussion during this time; table it, and take it offline for further discussion after the meeting. Remember that the team members aren't addressing you with a status update; rather, they are addressing the team about their work and their issues. There is no hierarchy in this meeting; no single person needs to lead it every day.

**Figure 6.1** A Team Conducting a Daily Scrum



Each member of the Agile Team is expected to answer three questions as part of the Daily Scrum (see Figure 6.2):

1. What did I do (and what did I complete) yesterday?
2. What am I planning to do (and committing to complete) today?
3. What impediments do I have that may block my success today?

**Figure 6.2** A Storyboard Illustrating Tasks Discussed during a Daily Scrum



Your leadership is crucial to the success of this daily meeting, even when other team members take turns facilitating it. You set the example for the rest of the team and show them that they can depend on you to remove impediments and grease the wheels of productivity. Your example should reveal the importance of this daily stand-up, and you should ask the right questions to elicit good communication and collaboration within the team. The specificity of your questions will reaffirm to the team that they are committing to work every day. Noncommittal behavior can easily arise from bland and vague updates. Remember that the purpose of this meeting is to align the team and commit to completing certain tasks on that day.

## Agile Team Questions

1. When is the best time to have this meeting?
2. Who is designated to lead the meeting and be responsible on different days?
3. What should be the core hours in which team members are available to collaborate with one another?

# Chapter 7

## Introducing the Product Owner, or Value Driver

The functions of Tribes vary: there is always a ton of work to be done ranging from analysis and research to building, hunting, and even warring. Because so many top priorities and competing interests exist in many different areas of business, we assign a Product Owner, or Value Driver, to a special leading role within the business. I often call these individuals *Value Drivers* because they fully and uniquely understand the priorities of different business units (and how the Agile Team must address them); they know where the highest value is at any given moment because they continually make decisions to ensure that the features or functions with the biggest benefit to the business receive primary consideration.

Product Owners (Value Drivers) work directly with Agile Teams, helping the members prioritize their workload, and also serve as intermediaries between the users and the Agile Teams. What needs to be built now? The Product Owners know, because they define project scopes and requirements. What needs to be built next? Product Owners know that too, because they prioritize and order the work. What does the team need to prepare for? Product Owners are already one step ahead in that they constantly groom the next set of requirements and priorities for their respective teams to consume. Product Owners also work directly with the stakeholders—those who give the Product Owner strategy. This is a one-person job, not a role to be filled by a group. While other individuals may have a say in the priority or value of a feature or function, they must go through the Product Owner, and the Product Owner makes the changes. The Product Owner's primary concern is the content of the product and the specifics entailed, while your Agile Team's expertise provides the implementation and development of the product.

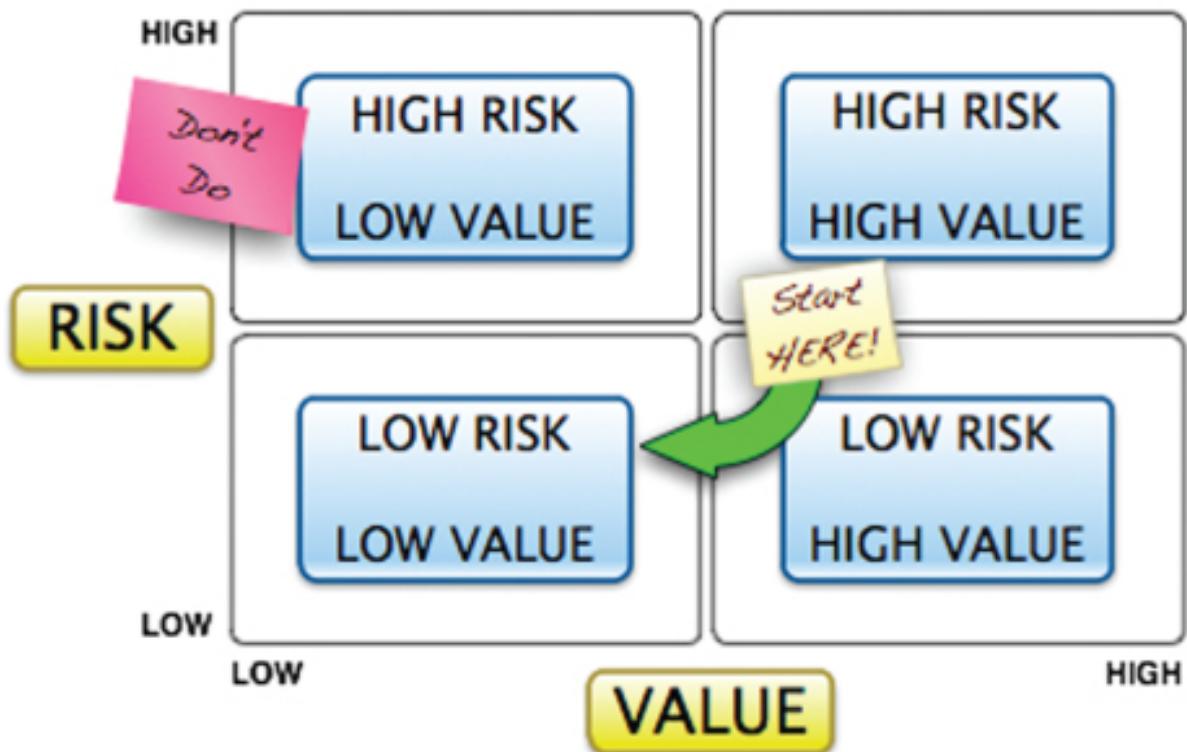
So, to begin moving toward facilitating the Scrum framework, your team must have a clearly defined Product Owner. This single person must possess the following characteristics:

- Intimate knowledge about priorities—an understanding of what the Agile Team needs to build or to do next.
- A high level of engagement with the Agile Team and a reasonable availability to help refine requirements.
- The ability to serve as a direct link between you and the stakeholders.
- Enough respect and empowerment to make value and priority decisions that the business needs.
- The ability to serve as the primary funnel and filter through which other conflicting priorities are settled.

The Product Owner (Value Driver) is a VIP because that single person knows the explicit and implicit value of the programs and projects that need to get done. The Product Owner gives the team unambiguous direction and allows the team to focus on that direction without having to worry about conflicting business priorities. The Product Owner is also the first line of business engagement for the Agile Team, providing support proactively and continuously to the team as a result of constant communication with customers and stakeholders about the details and priority of the product. The Product Owner is also available to each member of the team and is thus invaluable, ensuring that team members have the information they need to build the best product.

Remember, the Product Owner should prioritize the work by value. A value matrix can help your Product Owner determine what features should be built first. (See [Figure 7.1](#).) Start with the highest value features!

**Figure 7.1** A Value Matrix Determining Workflow Priority



## Agile Team Questions

1. Does your Product Owner have the qualifications necessary?
2. Does your Product Owner have the visibility and the empowerment to make priority decisions?
3. Does your Product Owner have the availability and engagement to provide the right guidance and course direction for the team?

# Chapter 8

## Discoveries from the Product Backlog

The Product Backlog is a list of all the tasks the Agile Team will work on for a particular project. After giving the orders of work and the priority of those orders, a Product Owner will walk to the Product Backlog and post them. All team members go to the Product Backlog for their work orders. This backlog is centrally located so that no one has an excuse for not knowing what to do, and it is ever-growing and changing for each project. (The Product Backlog is never complete.) At the beginning of a project, the Product Owner meets with the Agile Team and you, its Leader, to discuss the overall project and the highest-level details of what needs to be built.

Some teams have called this initial meeting a discovery meeting. Discovery meetings are usually pretty short, and the team doesn't spend too much time at them, although they may cover everything from the features and functions that will be built out for the project. One way to start building out the Product Backlog is to fill it up with as many requirements as a certain project may demand (see [Figure 8.1](#)) and then break them out into different *user stories*, or tasks (more on this in Chapter 9). As this list grows, you'll probably want to add some grouping mechanisms to organize the work. The use of *tags*, or group names, is a good way to start.

**Figure 8.1** A Packed Product Backlog Needing to be Completed for a Release

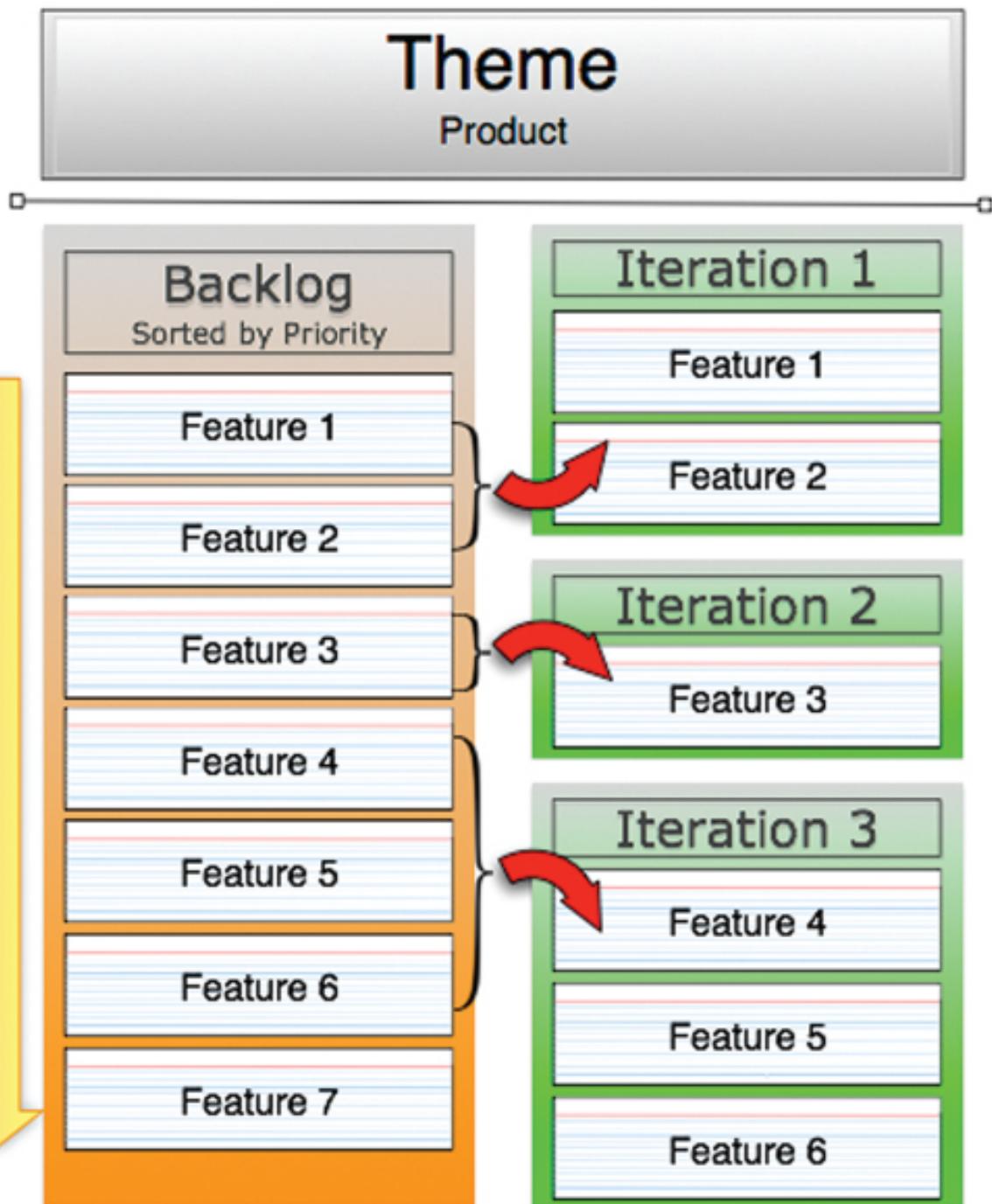


The Product Owner, who also owns the Product Backlog, must:

- Prioritize the backlog, placing the most important functionality at the top and the least at the bottom, according to business value.
- Be ready to consult and give guidance on any questions the team has about the product.
- Allow for any technical task to be inserted as a corollary to some of the tasks on the Product Backlog.
- Be able to give more detail to the higher priority items and articulate the value associated with the top items on the Product Backlog.

After the Product Backlog has been filled up with high-level stories, or functional units of work, the team is ready to estimate and task out work for a sprint. (See [Figure 8.2](#).) Your Scrum team is getting ready to build something fantastic. Get ready—the work is about to begin!

**Figure 8.2** A Backlog Prioritized Top to Bottom and Divided into Iterations



## Agile Team Questions

1. Has your Product Owner spent just enough time building out the requirements or foreseeable tasks to the Product Backlog?
2. Does your team have enough information from the Product Owner to begin work immediately?

**3.** Do you know of any impediments, constraints, or dependencies that the team needs to be advised of?

# Chapter 9

## The Sprint Backlog and Release Planning

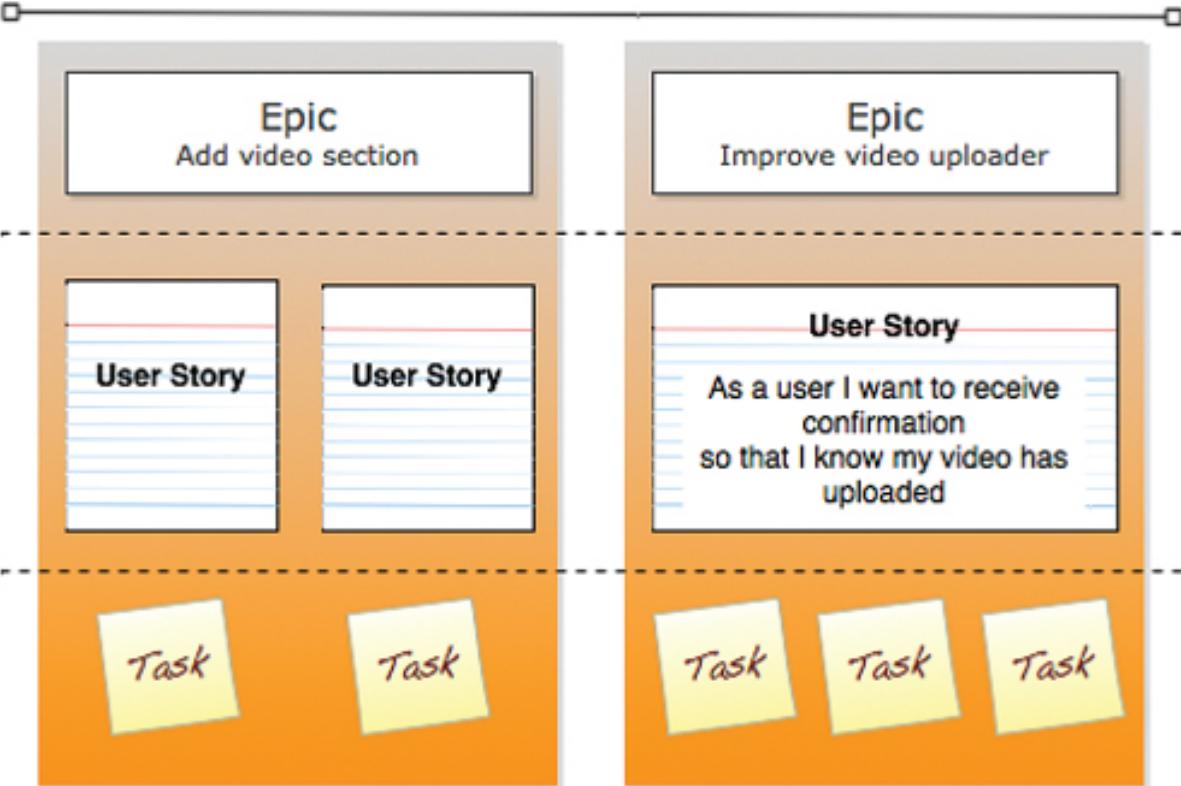
After the Product Owner has identified and prioritized all of the work items, the Agile Team then goes off to estimate the workload of the highest-priority items into a Sprint Backlog. The Sprint Backlog is the list of requirements or tasks that will be completed during a sprint, or set period of time. All of the items in the Sprint Backlog must be distilled enough for a team member to work on. Ideally these items will be broken out and sufficiently defined so that a team member can finish a piece of work within a day.

Everything in the jungle has a soul and a story—even project work. Since the Product Backlog holds all the items and functionality that needs to be built into a product, the Agile Team breaks down the work into user stories in a *backlog grooming session*. A backlog grooming session can take place at the beginning of a project, but should also continue constantly through the sprint cycle. This meeting is the meat and potatoes of the preliminary tasks that need to be done before work begins. The team members take as long as necessary to put flesh on the bare-bones tasks (called a grooming session). ([Figure 9.1](#) shows tasks broken down by theme.) This can take a couple hours while the entire team works in coordination to put all of the stories together to fit in the first sprint. (Sometimes a grooming session is combined with sprint planning, as described in Chapter 10.) This first sprint is a Timeboxed iteration. As teams progress and mature, I recommend that Product Owners and Agile Teams constantly take time out of a current sprint to groom the backlog and stories in preparation for the next sprint. This makes sprint planning shorter and allows the work to go more smoothly, and it has the potential to remove issues before the next sprint begins.

**Figure 9.1** A Theme Broken Down to the Task Level

# Theme

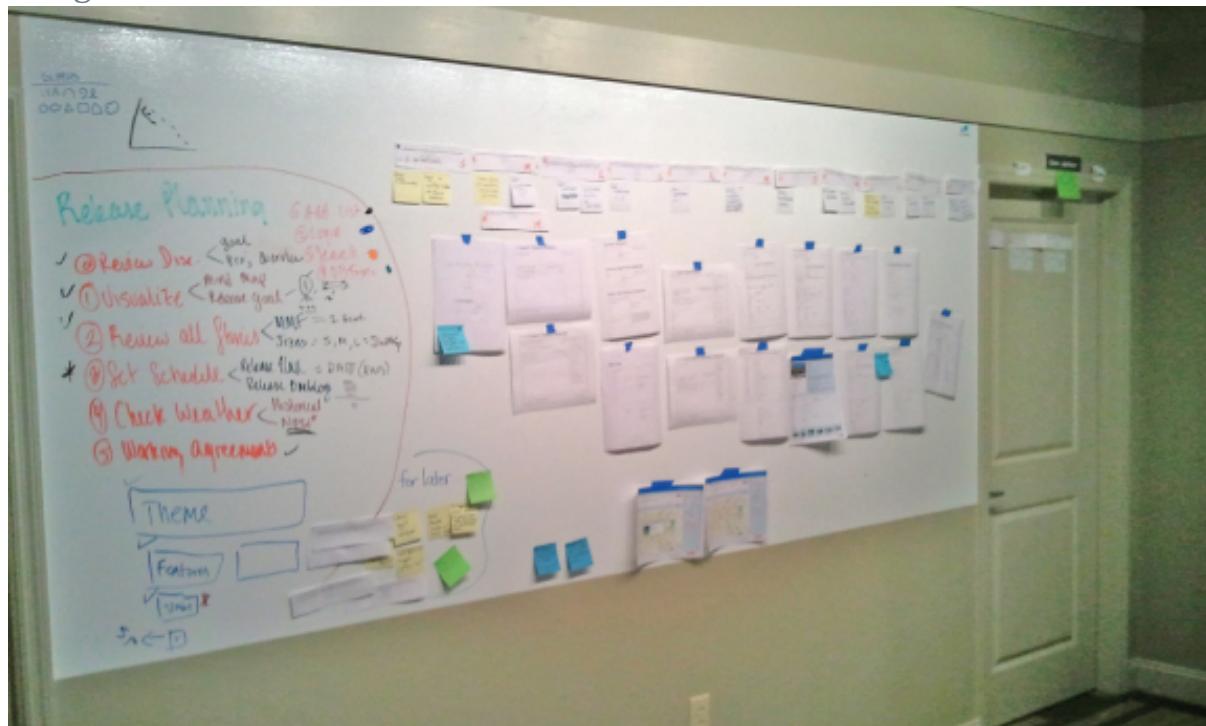
Add a video page to user profile pages



Many Agile Teams have found that two- or four-week Timebox iteration works very well in the beginning. Sometimes this Timeboxed sprint can move to one week! What can be necessary with larger projects, but is not always so for smaller projects, is a *release planning meeting*. Simply put, this meeting between the Product Owner and the team is to discuss, from a high-level view, the final deliverables, the team's initial capacity for work, and a high-level estimation of the approximate work that will be needed to complete the entire project. The release planning meeting helps when there is enough work to fill up multiple sprints, and the incremental development and deployment of the product needs to be tracked and managed. If your team determines that you need to have a release planning meeting for a larger project, then you'll want to make sure that you meet to discuss how many sprints (estimated) will be necessary to complete a product for deployment. Remember to record all the decisions, risks, and assumptions that come with the planning of the releases. The outcome of the release planning meeting is to build out a framework (or roadmap) and initial estimate for the delivery of the

product. (See [Figure 9.2](#).) This plan can and will be revised and changed as the project moves along.

**Figure 9.2** A Release Planning Roadmap with features at the Top, and UI designs and stories below



User stories are the backbone of your work. The stories:

- Must be posted in a highly visible area for all to see.
- Must be updated daily.
- Must be as descriptive as possible.
- Must be evaluated and estimated for work (more on this in Chapter 10).
- Must be estimated by the team and maintained by you.
- Must be sized correctly to fit in a sprint (more on this in Chapter 11).
- Must come from the Product Backlog.
- Can be added to, removed, or changed, but only by the team.

The Sprint Backlog can be maintained in several different ways.

Remember, the Sprint Backlog is the list of items to be worked on for the time-boxed iteration. Most often, Agile Teams have found that creating a wall that tracks every user story has been successful. It is absolutely imperative that the team and you maintain the items in the Sprint Backlog, as your team will be committing to the completion of the tasks as they are broken down into sprints or iterations.

## **Leader Questions**

- 1.** Where can the user stories be located centrally for the Agile Team to discuss? Note: You can also use software to track your stories, but many teams have found that a physical wall has been of greatest value (more on this in Chapter 13).
- 2.** Does each project require a release planning meeting?
- 3.** Has everyone on the team participated in the meetings and understood the planned approach to the release of the product?

# Chapter 10

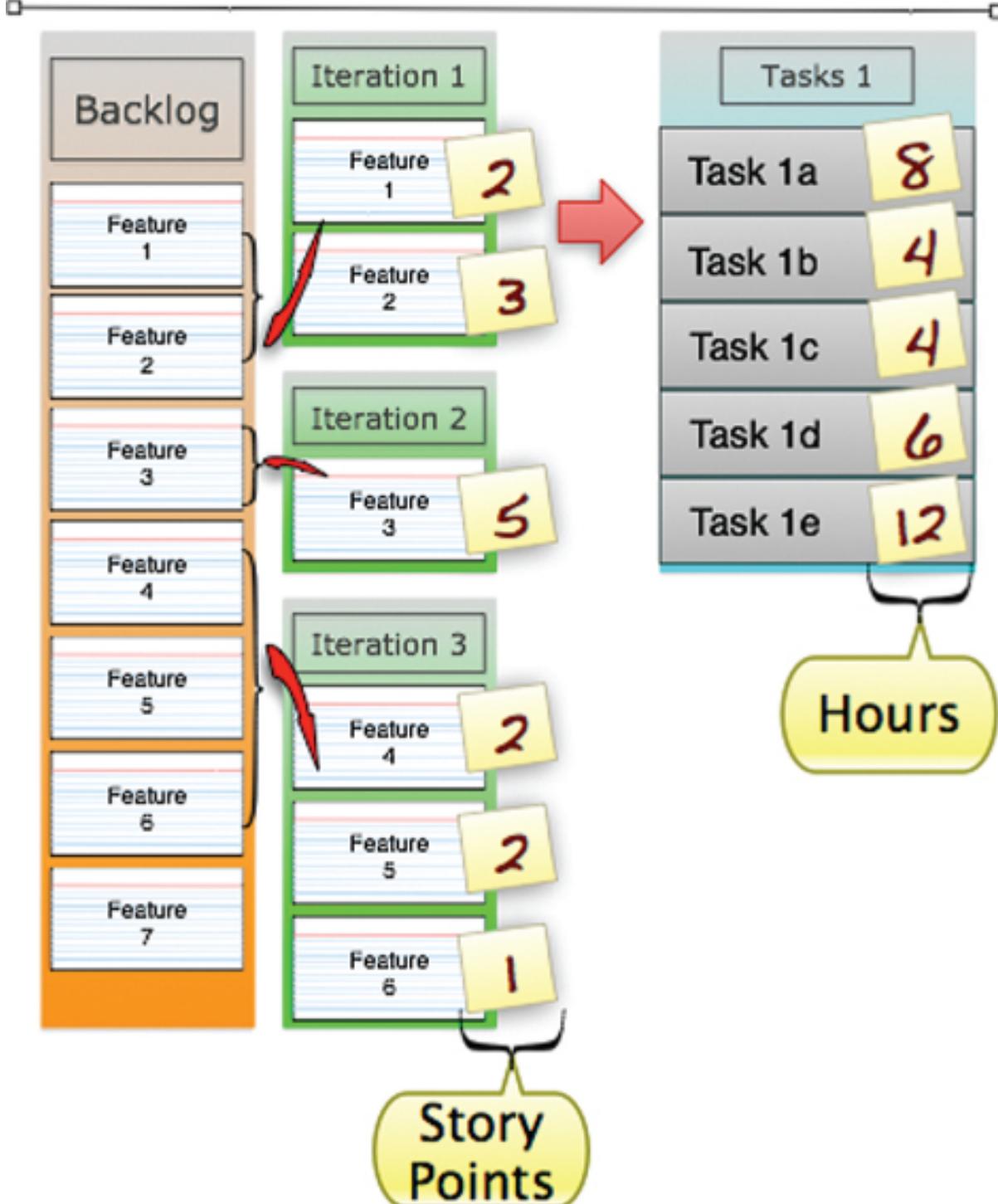
## Sprint Planning Meeting

The Product Owner and you are now ready to set out the schedule of work for the upcoming iterations. The sprint planning meeting is the perfect place to start setting the pace for work. Remember, the team has a good grasp on the vision and purpose of the project, and you are now ready to commit to stories and the work at hand. Now, the Product Owner, you, the entire team, and any other interested members of the business work together to understand how many user stories they can commit to completing within a given sprint. These sprints hold a certain point value, as we discuss in Chapter 11. At this point the team answers the questions, “What needs to be built first, and can we commit to completing it?” (See [Figure 10.1](#).) These user stories are generally locked in place and committed to by the team. The Product Owner can change some stories and requirements as long as the team commits to completing them or you help the Product Owner understand that some features may be added to the sprint, which may mean that other features will be removed. We then estimate the work as detailed in Chapter 11. It's “go” time. Let's get started.

**Figure 10.1** Sizing Stories with Story Points into Tasks with Task Hours

# Theme

## Product



Note: Many Agile coaches do not have a backlog grooming session prior to a first sprint for a new Agile Team as depicted in Chapter 9. That is absolutely fine! I have found that it is wise to allow the team to learn the

process and get a good grasp on breaking requirements into user stories, tasking, estimating, and aligning themselves to a goal. As teams mature, your teams will fly through a sprint planning meeting because you've been grooming the backlog with the Product Owner throughout the week in preparation for the sprint planning meeting.

The action items that result from a sprint planning meeting include identification of:

- The sprint goal or theme.
- All items of value that the team intends to deliver.
- The highest-priority value items first.
- The tasks needed to complete each story.
- An agreement process to evaluate any story additions/removals from the sprint.
- How much buffer should be included in the work (e.g., meetings, other responsibilities, ad hoc changes, etc.).
- The commitment by the team to complete all of the assigned stories on time.
- The estimates of the user stories and tasks (more in Chapter 11).
- The Timeboxed length of the sprint (determined initially; from then on it should be a standard amount of time).

From the outpourings of this meeting come a Sprint Plan, or a measurable way to determine the success of the project as well as to ensure that everything committed to in the Sprint Backlog reflects the product or project value. The success of the sprint will later be assessed during a retrospective meeting (covered in Chapter 15).

## **Leader Questions**

1. Has everyone, especially the Product Owner, participated in the meeting?
2. Has the entire team committed to a plan that is realistic and achievable?
3. Has the Product Owner given the team the most up-to-date information and features as well as an organized and well-understood list of priorities?

# Chapter 11

## User Stories and Estimation

Each and every Tribe and business pass down stories from generation to generation. These stories tell of past projects and teachings that the Tribe has learned and grown from. The best of these stories are the ones rich with content, detailed in their specifics and complete in their telling.

Each endeavor for the Tribe is a conglomeration of stories; the more depth these stories have, the easier they are to know, understand, and apply to the project. In total, the stories need to be analyzed and understood by the Agile Team. What comes out of user story writing is the desired functionality of each unit of work for the project. These user stories are often written on index cards or sticky notes. During backlog grooming or sprint planning, the team members will assign themselves stories and commit to the work. Their signature on each user story commits them to the completion of the work. (See [Figure 11.1](#).)

**Figure 11.1** Team Members Assigning Stories and Committing to Work



Each user story created around a particular feature or functionality always follows the laws of the land in user story creation. (See [Figure 11.2.](#))

- As a <user> I want to <function> so that I can <business value>.
- Acceptance Criteria (AC): <Steps necessary for this story to succeed or be completed>.
- Notes: <Any additional information pertinent for the developer>.
- Priority: <If you have a priority scheme>.
- Effort: <Unit of effort to complete story>.

**Figure 11.2** A Story Card Written on a Sticky Note

## ENABLE ATTACHMENTS

#SZ.12

as a user  
I want to  
create a button  
so that the user  
can upload attachment.

-AC: Button made  
function created.

Here is an example of a story:

*As a <customer> I want to <build a drop-down box on my website>  
so that <I can easily see a selection of inventory that I can buy>.*

The story's AC includes these five points:

1. Drop-down box needs to be gray.
2. Drop-down box needs to be 30 pixels in length.
3. When clicked, drop-down box must show five items in the beginning with the rest below through a scrollbar.
4. Items in the drop-down box must be in alphabetical order.
5. Items must also be sorted by size.

Notes: This drop-down box must also include new products we are adding to our inventory in one week. This box is custom created for a top-dollar client.

Priority: 1

Effort: 4

Backlog grooming sessions or sprint planning meetings are also the place where each user story is estimated for level of effort. This is a

measurement of complexity of work, not the specific time to complete it. Since we all know we're terrible at estimation, how do we do this?

In estimating stories and work, remember to make sure to:

- Use relative units (e.g., A is half as hard as B).
- Take into account team member skill level, domain knowledge, external interruptions, and unexpected technical issues.
- Solicit group input on the estimations.
- Have everyone estimate in the same units and agree what “done” means.
- Do not estimate when your team is unsure about the specifics of the project.
- Create reference stories that the team can use as a template (or templates) for estimation.
- Be realistic.

There are certain ways of estimating stories. Often teams have found the use of doubles as a great way to estimate (e.g., 1, 2, 4, 8) or the Fibonacci sequence (e.g., 1, 2, 3, 5, 8). With many Agile Teams, any type of estimation of an “8” is considered an “epic” and needs to be broken down into smaller pieces that can be estimated as a 1, 2, 3, 4, or 5 depending on chosen estimation method. These points end up being the velocity that you will use to measure the total amount of work or capacity your team can handle in any given iteration. These points need to take into account external interruptions, technical surprises, team member skill level, domain knowledge, and other factors. Over time, this velocity will need to be stable to accurately measure the team's capacity. The most-used graph for this is a Burndown Chart, which graphs total velocity points over iteration time (more on that in Chapter 13).

The quickest way to estimate user stories is through affinity estimation. (See [Figure 11.3](#).)

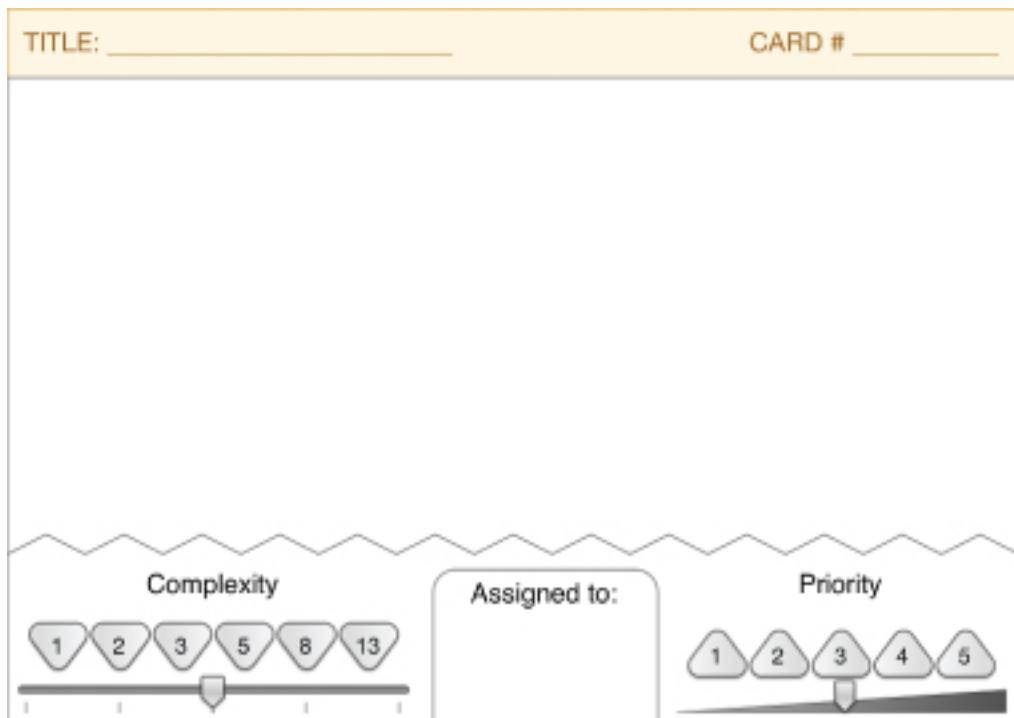
**Figure 11.3** A Team Doing Affinity Estimation



Affinity estimation is far better than static guessing of complexity. We are far better at comparing than static estimation, so we make use of our natural gift of comparison to get estimates quickly.

- Put number buckets up on a wall using sticky notes: 1, 2, 3, 5, 8, 13 (Fibonacci). Smallest and least complex user stories on the left, and largest and most complex on the right.
- The team discusses stories to make sense of them.
- Utilize a baseline story (for a first Sprint), often this is the easiest user story for the team to complete.
- Without conversation, the entire team takes turns moving stories under each numerical bucket as to how complex that story is to complete, also comparing each story to previous stories. (See [Figure 11.4](#).)
- Anyone can move any story position.
- As activity subsides, collaborate, refine, and then prioritize.
- Task out the prioritized stories and commit!

**Figure 11.4** A Story Card Template



Planning Poker is another way of estimating stories. You can pick up a deck of Planning Poker cards at <http://store.mountaingoatsoftware.com/>. Essentially the game plays as follows:

- Each developer has a set of planning cards.
- As each story is brought to the group, all developers flip over an estimation card for the total work to complete the story.
- If all of the estimates are similar, then that is the assigned value.
- If there is a discrepancy between developers' estimates, then a conversation ensues as to why and what the more accurate estimate is.

## Agile Team Questions

1. Who can be responsible to lead the sprint planning or grooming and continue to communicate and uphold the standards of user story creation?
2. What type of estimation point system should we employ?
3. What should be the criteria for complexity when estimating?

Here are some suggestions from authorities in the business about ways to create a good user story:

## Ron Jeffries's 3 Cs<sup>1</sup>

1. Card—each user story is represented by a card or token.

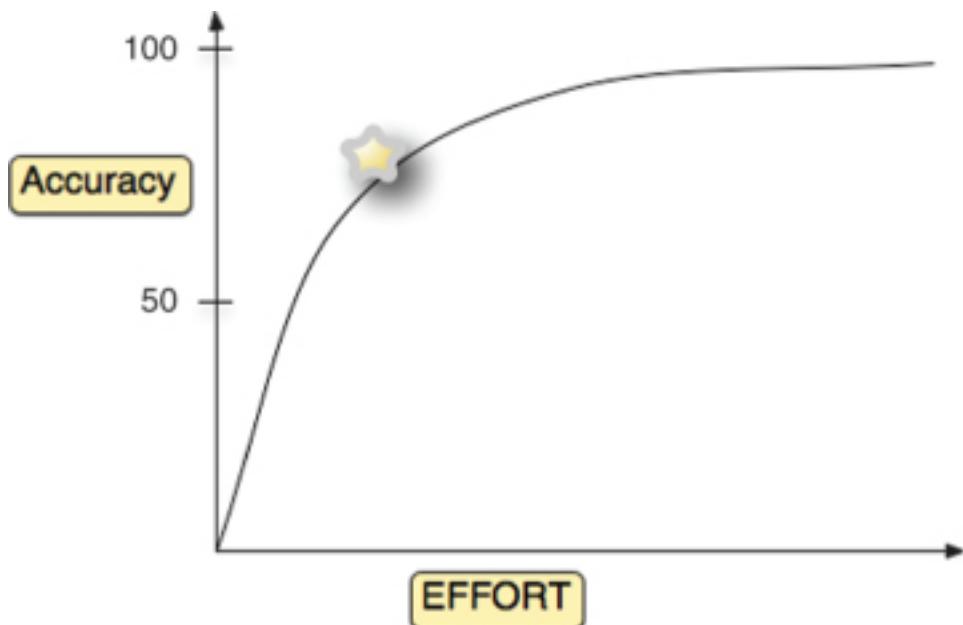
- Conversation—the conversation around that requirement and user story is the most important.
- Confirmation—confirmation of understanding (and confirmation of acceptance criteria) ensure that the user story will be completed according to specifications.

## Bill Wake's INVEST Model<sup>2</sup> regarding what a story card should be

- Independent—each user story should be able to be completed independently, without dependence on another user story.
- Negotiable—each user story should be negotiable around priority, value, and specific requirements necessary for completion.
- Valuable—each user story should have an explicit value to the business.
- Estimable—each user story should be broken down enough to be estimated.
- Small—each user story should be as small as possible so a team can consume and complete user stories quickly.
- Testable—each user story must be able to go through all testing requirements and procedures.

Whatever story method you use, or whatever advice you take, remember to not spend excess time estimating! (See [Figure 11.5](#).) Spend just enough time so that you get the right balance of accuracy for amount of effort.

**Figure 11.5** Balancing Accuracy and Effort



- <sup>1</sup> Ron Jeffries. “Essential XP: Card, Conversation, Confirmation.” [Xprogramming.com/articles/expcardconversationconfirmation/](http://Xprogramming.com/articles/expcardconversationconfirmation/)  
<sup>2</sup> Bill Wake, “INVEST in Good Stories and SMART Tasks,” <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>.

# Timeboxed Sprints (Iterations) and the Meaning of Done

Your Agile Team wants to be as productive as possible, but not only that—your team wants to optimize efficiency, increase predictability, and reduce risk. The various other business units are vying for your attention and resources at this moment, and the leaders from those Tribes want to see results as soon as possible. They have almost stopped going to the mystical waterfall and the processes that it instills in people. They are waiting on you to report back to them that work can be done faster than a 9–12 month plan. You have your marching orders. You have a project plan. Now it's time to break those out into logical time units where you can complete working product. Figure 12.1 shows a step toward the goal: product release.

**Figure 12.1** A Release and Sprint Calendar

## Definition of Done Brainstorm



When you have spent time estimating the tasks and features, gather your team and all members working on the particular project. It is now time for them to help you commit to a certain amount of work within a given time frame, and now is the time when using an online Agile tool can help.

Timeboxed sprints need to have the following guidelines:

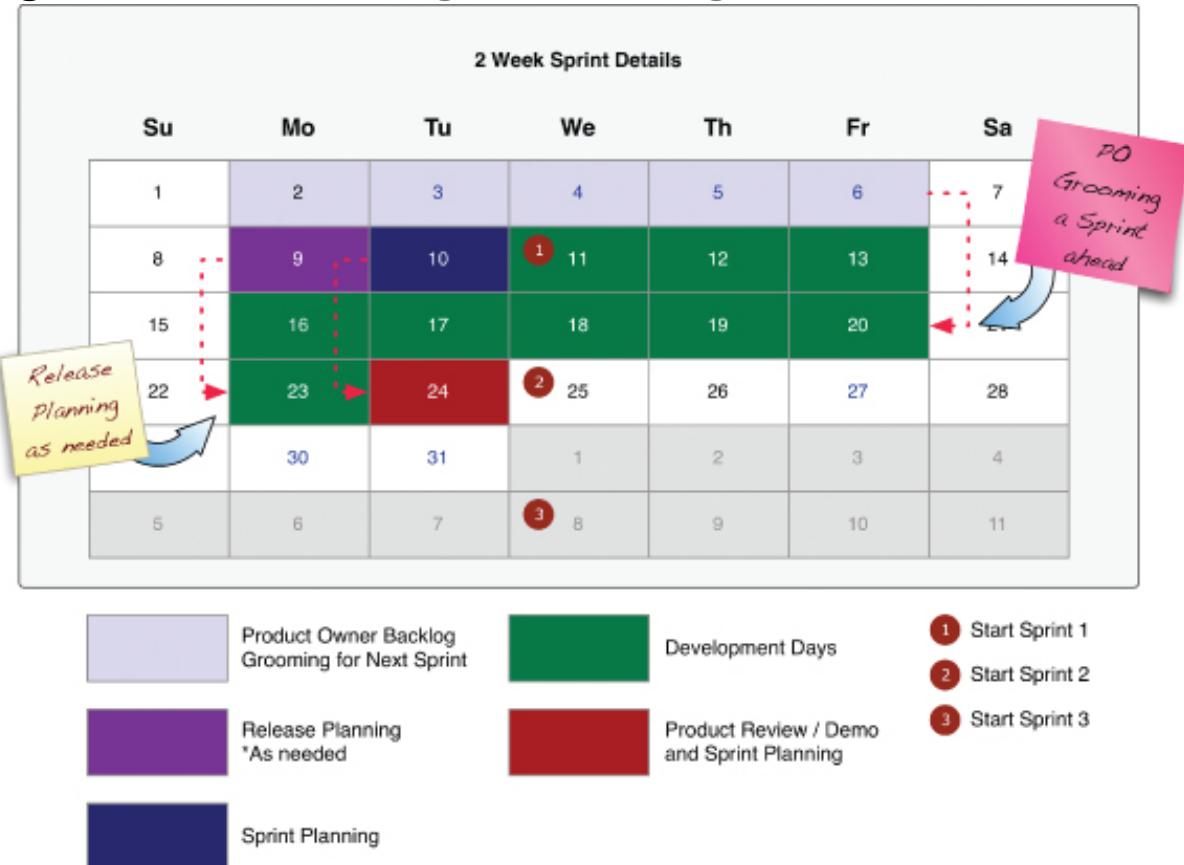
- The sprint goal is established and well understood throughout the team.
- All sprints are four weeks or fewer. (Many teams have found that they can successfully push product out even in a week!)
- All sprints must end on time.
- All sprints must have a team that is neither interrupted nor controlled by other stakeholders or other business tribes.
- All sprints will complete with almost everything they have committed to. (As your team becomes more accurate, they will be able to complete 100 percent of their features nearly 100 percent of the time.)
- Sprints can be canceled, but only by the Product Owner and/or you, and if there are business reasons for doing so. (Most likely this will not be necessary due to the short length of time the iterations take up.)
- The shorter your Timeboxed sprints are, the shorter your estimates will be for tasks.

Your Agile Team enjoys the ability to have a lot of wins weekly. This has completely improved your team's culture and attitude at work. This boost in morale is a welcome addition to your team's working environment. Tread carefully here, though; many teams can lack maturity and experience working under these guidelines. That's okay—it is a learning process! When work doesn't get done within a Timeboxed sprint due to overcommitting or underestimating, it is time to gather the team and reflect on how things may be improved. It may mean that the sprints need to be a different length, such as three weeks instead of two. Or it may mean that they need to split the bigger user stories down into smaller user stories. Or, it may mean that the team needs to reflect on how they develop, test, or deploy their code. Take the time necessary to define areas of improvement; these conversations with your team will bring them back up to speed and not keep them in a death-spiral of despair. Be a leader and energize them to work smarter together. Your attitude here is crucial to the success of achieving consistent sprints.

One of the major definitions your team has to understand is the meaning of *done*. Business stakeholders, developers, quality assurance personnel, business analysts, and others may assert that *done* means something completely different than it does to others. Regardless of how many definitions are out there, your team must establish a definition of *done* that the Product Owner and team understand clearly. As the Product Owner is clearly the client, his definition of *done* is king. As your

team sets out to complete the sprint goal, remember to clearly define what *done* is for your Product Owner. Does it include all requirements, testing, documentation, security, and integration? Or is it something simpler? Define it! (See [Figure 12.2](#).)

**Figure 12.2** A Brainstorming Session Defining *Done*



## Agile Team Questions

1. What is the best Timeboxed sprint length that your team should employ?
2. What is your team's definition of *done*?
3. What is the defined functionality of a completed iteration?

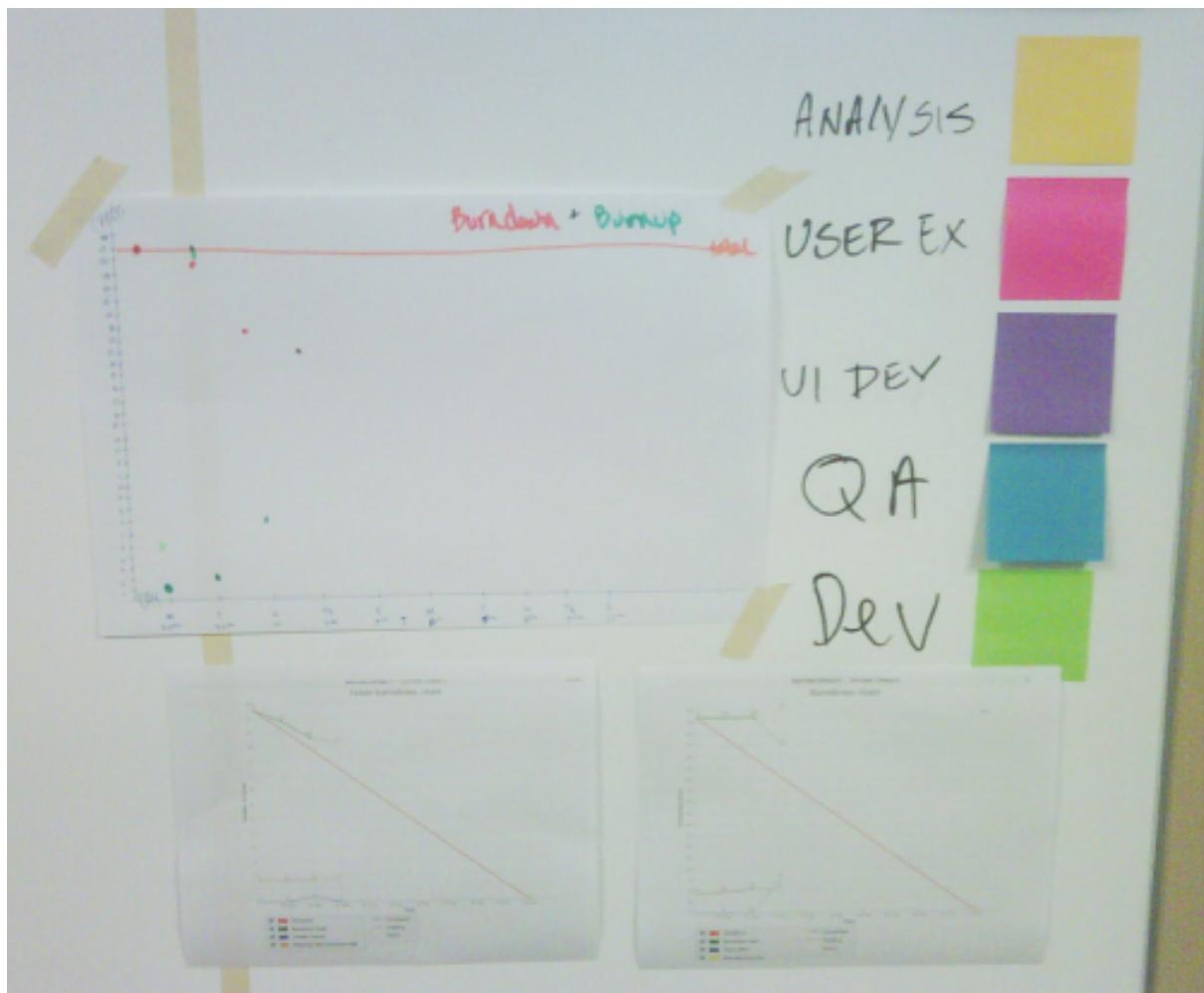
# Chapter 13

## Tracking Flow and Information Radiators

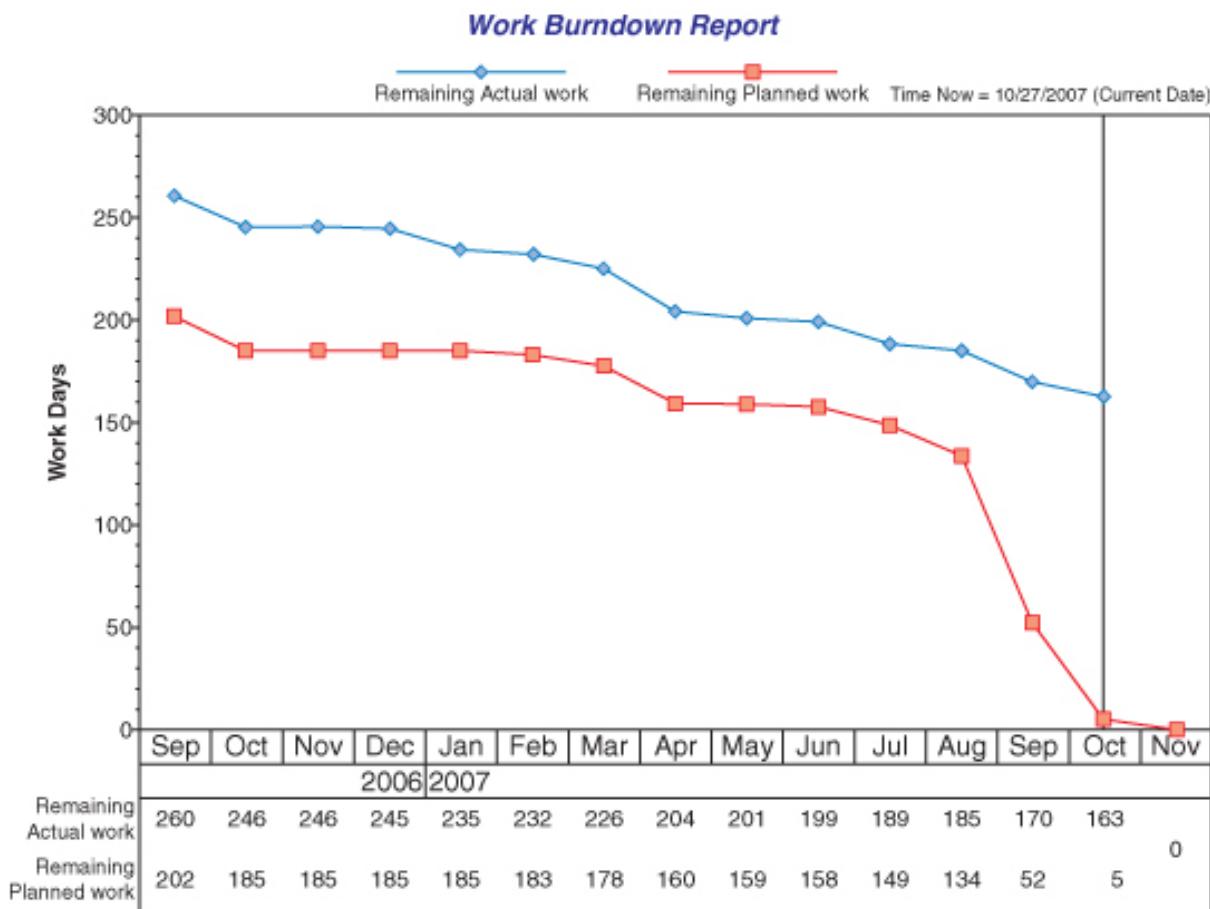
You now have a backlog full of work items, user stories that have been broken out into sprints, and a plan to get all of the features done within a certain time frame. Your team is ready to go. All you need now is a way to track progress through the sprints. What you need now is a highly visible area in which you can build your swim lanes of progress.

Your Agile Team's backlog has all the user stories, possibly hung on a wall somewhere. How do you track when the user stories get worked on? A simple process could be to start with three different swim lanes up on a wall: *backlog*, *working*, and *done*. Let's say a team member begins work on a user story; she moves it to the working section, and when she is complete, she moves it to the done swim lane. This simple flow is a great starting point for your team. Each team is different, and some enjoy breaking out swim lanes into very specific buckets. See what works best for your team. The charts discussed in this chapter, which can also be called information radiators, provide examples of what other Agile Teams have done to track progress and flow. (See [Figures 13.1](#) and [13.2](#).)

**Figure 13.1** Burndown Charts Make Useful Information Radiators



**Figure 13.2** A Burndown Chart Tracking Workdays over Months



You have a great idea—two great ideas, in fact. They're called a Burndown Chart and a Release Burndown Chart. These charts enable your team to track daily progress as it is completed. The *x*-axis of the Burndown Chart is your sprint length (in workdays) and the *y*-axis is total number of user story points to be completed during the sprint. For a Release Burndown Chart, the *x*-axis is the total number of sprints and the *y*-axis is the total number of user story points for the entire project. Both of these charts have a line that shows the predicted completion of user stories over time. These are very simple ways for you to track the progress of the work being done by your team, and they give your team a simplistic view forecasting when the work will be complete.

What happens when work is added midstream or scope is increased? What you will see is a spike in the chart! A regular Burndown Chart won't show you the details if issues arise. Another chart your team likes to use is a Burnup Chart. This chart is updated daily and shows the total work over the period of time; it also includes a line at the top to show whether additional work has been added or removed. When there are changes to the number of user stories within a sprint or iteration, it is recorded in this chart. Personally, I think a Burnup Chart has more value to the team. It allows the team to see how scope changes can affect

timing and whether they are able to adapt to changes on the fly. As user stories are completed, the line goes up toward the total number of story points.

Here are some points to consider when using Burndown or Burnup Charts.

- Charts are simply conversation starters; they should reinforce the need to discuss any deviations or notable changes in work through a sprint or release.
- Charts need to use the team's velocity as a gauge for workload. Do not track hours over time, as that isn't very agile! Estimation points or story points created during the sprint planning meetings are best used. In due time, the team should be able to get to a consistent number or average number of points per sprint.
- Charts allow you to measure the rate of delivering working software to your stakeholders. This velocity should be your main metric in determining work capacity and value.

Tracking the flow and the progress of your team's work is crucial for transparency to both the team and the business. These charts, or information radiators, allow disclosure to be part of the team culture as well as the business culture. Ideally, the charts will allow the team to develop in time a predictable pattern of work and stabilize as the team learns what works and what doesn't. Remember to update all flows and charts daily. Even have your Daily Scrum in front of the information radiators to allow your team to speak to specific user stories or tasks.

## **Agile Team Questions**

1. Where is the best place to put these information radiators in your office?
2. What trends does your team see over time?
3. Over time, how much work can your team handle within a sprint?

# Chapter 14

## Demonstration of the Product

The Tribe is getting excited about putting structure into place where there was none previously. It is apparent that after each sprint there needs to be a demonstration or review of the product so that the Product Owner and business can see the positive output of the Agile Team. This is a great opportunity to get some early feedback from the Product Owner about what is working and acceptable in the product and what isn't. This feedback can help the team during the next sprints to determine what needs to change from the previous sprint. The demonstration of the product needs to showcase a fully functional piece of the product that the team considers reviewable and acceptable by the Product Owner. In software-related environments, the demonstration proves that functionality is potentially shippable, or ready to be put into production. While the demo may be in a development or quality-assurance environment and all functionality may not be accessible, the software needs to be demonstrated as much as the capabilities of the environment allow.

During this demo, you'll want to make sure that the Product Owner is in the driver's seat. You may feel that a developer should show off all of the features, but through experience you'll find that the software sometimes acts differently and may behave in odd ways when someone other than a developer drives. You'll want to make sure that you navigate the meeting so that unnecessary distractions won't stall the completion of the demo. People participating in the demo may have great ideas, but if they aren't part of the agreed-upon features, then take notes and document the comments.

Demos and reviews are crucial because they:

- Happen after every sprint.
- Show working valuable product or tested software.
- Enable feedback from the stakeholders and Product Owner to be given to the Agile Team.

- Allow the business the opportunity to sign off on the product (acceptance).

A demo is a time for your Agile Team to shine (See [Figure 14.1](#)). Let the team share the features and give context around what the product is if the Product Owner has any questions or needs clarity. Quite possibly, you could demonstrate every user story. Let the enjoyment for what has been built come from the inside of the team, not you. Allow the team to revel in this opportunity to show what they have been working so very hard to complete—and on time! This should be an exciting event; let that excitement continue to grow and confidence be built as your team accumulates more and more successful sprints.

**Figure 14.1** A Team Demonstrating Features during a Sprint Review Meeting



The primary goal for your team will be the delivery of the product on time and within user story specifications. This focus allows the team to rally around a definitive goal, and the Product Owner will begin to get more and more involved during the process.

## Leader Questions

1. Are all of the necessary attendees available for the demo?

**2.** What is the best way to promote the announcement of the product to the other business units (e.g., an e-mail announcement)?

**3.** What ways can you reward successfully completed, tested, and working products or software?

Note: I've found that a couple of problems often come up when first instilling a process to demo a product. Regardless, you should always strive to have a demonstration of the product because it can be the single best point during a sprint where hard work is acknowledged and rewarded and the necessary feedback from the stakeholders is documented.

Here are three main issues that might arise when having a product demo:

**1.** *Functionality cannot always be demonstrated.* Still have a demo. You can use this time to demonstrate nontangible functionality, wireframes, and design compositions, even architecture. Use this opportunity to get feedback from the Product Owner.

**2.** *Developers do not have the time to do a demonstration.* Allow your team to build in demo time during each sprint or remove a user story or two so that time can be allotted for demos.

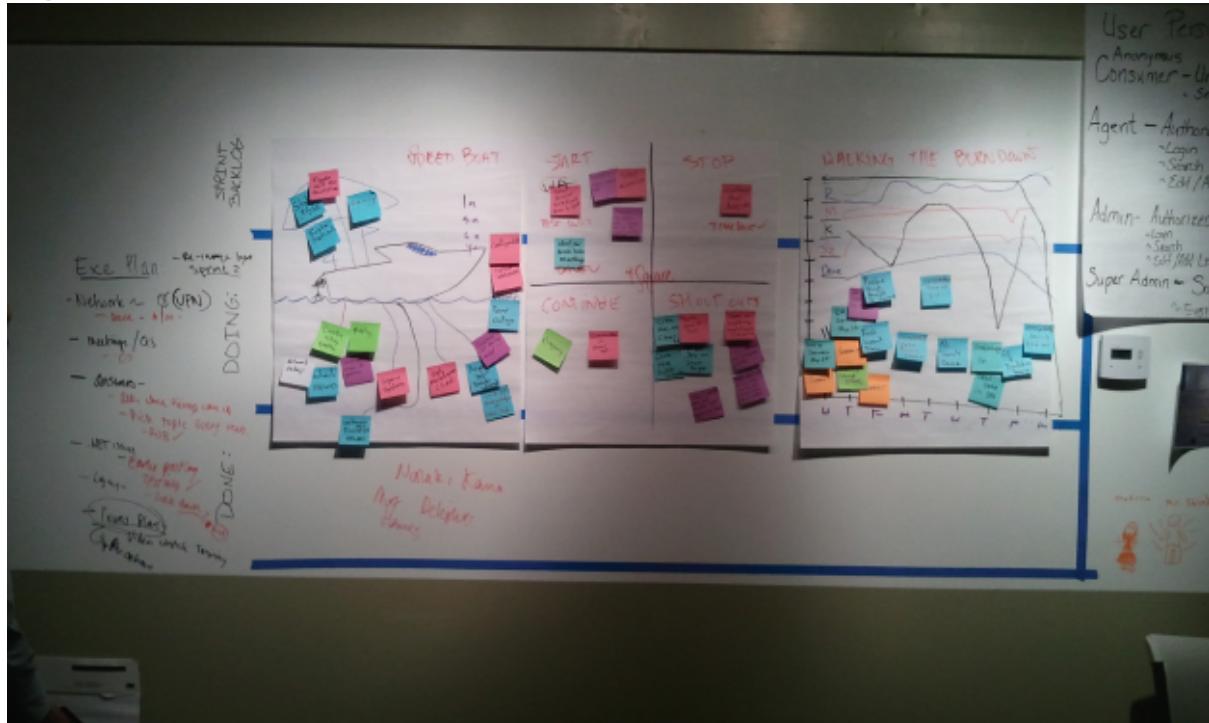
**3.** *Stakeholders and the Product Owner do not attend the demo.* Try scheduling a standing meeting that allows consistency for all parties to attend at the same time and same day for every iteration demo

# Chapter 15

## The Retrospective

Team members can be brutally honest. Their honesty needs an outlet. Team members can also be hesitant to share. Their input is valuable, and we need it. A retrospective meeting is a perfect place to enable your entire team to talk about the latest sprint. It is a moment for the team to take a breath and reflect, to find things that worked well, things that needed improvement, and areas of opportunity to improve for the next sprint ([Figure 15.1](#)). Everyone must participate and answer the questions on the agenda.

**Figure 15.1** Three Retrospective Workshop Exercises



Often the retrospective can be put on the back burner and ignored at the end of a sprint. Do not fail to have a retrospective. Put in a user story card during the iteration to set out time for this meeting if you have to. This retrospective allows your team members to build confidence in one another, inspect the process, adapt, and give you, their Leader, an

opportunity to motivate the team and receive feedback. (See [Figure 15.2](#).)

**Figure 15.2** The Leader Motivates the Team during a Retrospective



The attributes of the retrospective meeting include the following:

- It happens after every sprint.
- It includes the entire Team and the Product Owner.
- It is a safe haven for feedback of any kind (as long as it's tactful and constructive).
- It needs the participation of every single person.
- It needs a good facilitator who can stay out of the conversation and maintain the flow of ideas and thoughts.
- It can be a place where other activities are included. Mix it up!

Some Agile Teams have found that appreciating other people and giving the good news first is a great way to put your team in a positive frame of mind. This sets them up to take better advantage of opportunities for improvement. Again, honesty is paramount. Your team's ability to improve and see continued success is marked by its members' willingness to critique and challenge one another. Focusing on solutions rather than just the problems can help your Agile Team think as a team and encourage one another toward excellence.

Make sure to check out opportunities to create a retrospective workshop. You can find plenty of exercises online at several websites,

including [www.tastycupcakes.org](http://www.tastycupcakes.org), [www.gogamestorm.com](http://www.gogamestorm.com), and [www.innovationgames.com](http://www.innovationgames.com).

## **Agile Team Questions**

- 1.** What happened during the last iteration?
- 2.** What went well, and what can we celebrate as a success?
- 3.** What are things that didn't go so well, and how can we improve the next sprint?

# Chapter 16

## Wash, Rinse, Repeat, Win!

The sounds of celebration are wafting through the jungle. The melodic beats of the drums rise and fall as if the jungle is alive with one heart and one breath. The leaders in the business are popping their heads out of their windows and doors, all in the direction of your Agile Team.

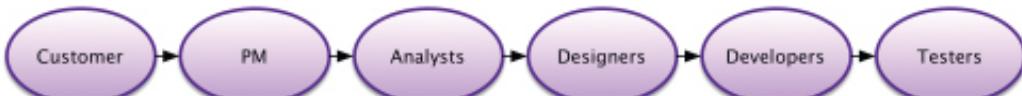
Something is brewing in the jungle, and an atmosphere of healthy anxiety is permeating the landscape like a thin fog. Early reports say that the Agile Team now has all the elements necessary for a successful rollout of Scrum, and some of the Tribes in the business are getting excited about the possibility of productivity increases and better communication and collaboration, as well as a unification of production throughout the different Tribes in the jungle. The gears of change are meshing, and it all began with you.

Agile management can be as simple or complex as you desire. Some things to think about include the following:

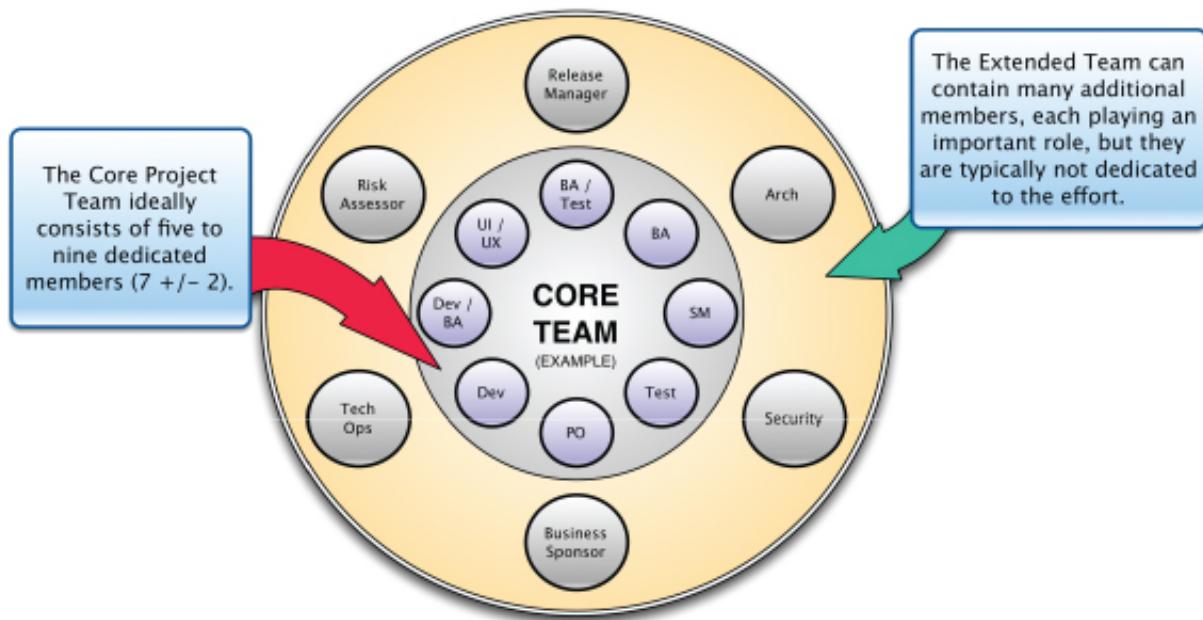
- Define as a Team which particulars of an Agile framework will be implemented into your culture.
- Define rules, processes, procedures, guidelines, and standards for your teams (i.e., standard meeting guidelines, estimation standards, communication channels).
- Define project members' roles and responsibilities (i.e., who is responsible for what). (See [Figure 16.1](#).)
- Define goals and plans for measuring the success of a project (i.e., what makes a product rollout successful).
- Define and promote iteration standards that are necessary for your company's delivery cycle (i.e., discovery, backlog grooming, Daily Scrum).
- Define a communication strategy of releases to all necessary business units.
- Define and understand your company's culture (see Chapter 17).

**Figure 16.1** An Agile Team Diagrammed by Roles and Responsibilities

Traditional Silos and Roles



## An Agile Team



In the continual effort for you to implement Agile processes in your organization, you will need to be cognizant of being absolutely consistent with your communication to all the different business units. Agile and Scrum in itself do not tell us how to do software and product development or prescribe processes for business analysis or product management. Agile is not a project management tool. It is simply a framework that can fit around the specific nuances of your organization. With that in mind, a healthy balance of transparency and disclosure will garner trust and willingness from the other business leaders as you grow your influence and success with Agile within the already established processes that your business has. Your best weapon is continuing to read and learn. Your best defense against being ineffective in Agile is a direct result of how much you continually educate yourself and arm yourself with the knowledge that comes from others who are far more wise and successful than yourself. Remember, this book isn't prescriptive, and it does not have all the answers! Your experience, skill, and common sense will augment many of the suggestions I've made. My hope is that this will be a great starting point for increasing agility, value, and fun at your company or team!

## **Leader Questions**

- 1.** Am I willing to set the bar high and set goals for my team and my company?
- 2.** Am I willing to be a patient Servant Leader who will do what it takes to be successful?
- 3.** Am I willing to be empathetic and show humility as a fellow learner?

# Chapter 17

## Team and Business Cultural Dynamics—Team Science™

Business leaders, executives, and managers must understand that team and business culture affect productivity, sustainability, and employee satisfaction. We are no longer in the manufacturing age where each employee is simply a cog in a big wheel. As a Leader, you hired people for their problem-solving abilities, technical aptitude, and fit; not for their good looks (I hope).

You want to ensure that each team has been enabled, equipped, and designed optimally. You want to ensure that each team member communicates, collaborates, and works well with others. Finally, you want to make sure that all impediments to productivity and cultural dysfunctions are removed.

How is it possible to do all this and create high-performing teams? The best way that I've found is through a cultural and team assessment. As a business transformation coach and manager, I want to know where there are gaps within team collaboration and communication. I want to understand how to engage intentionally with each team member and how to optimize each team member's potential, or strengths. When I say that I want companies to hire the right candidates and have the right levers to pull when building new or augmenting existing teams, I am not talking about resumes and recommendations. I want to definitively know whether candidates are exactly the right cultural fit for my company and team.

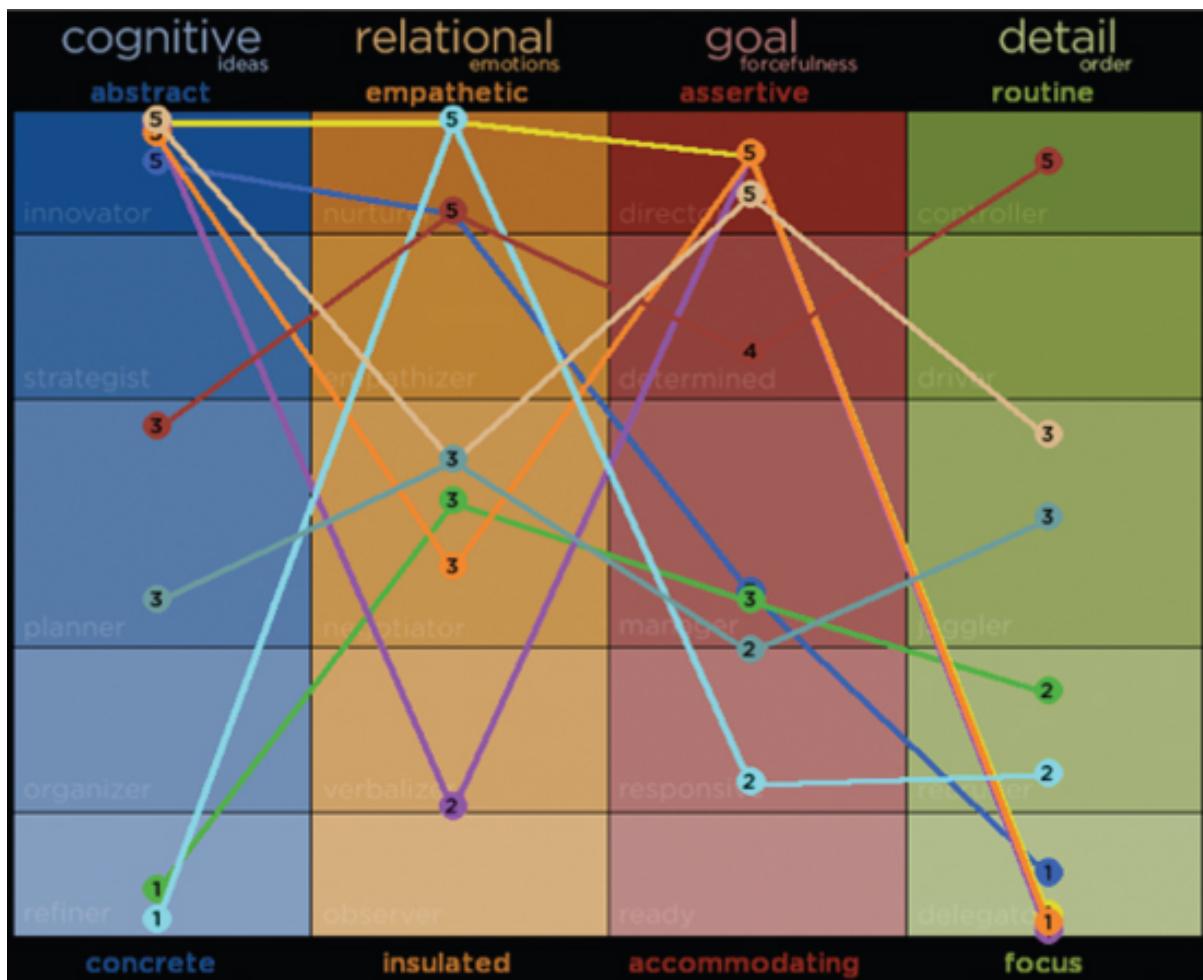
I've used several assessments in the past that have fallen flat on their faces. The most effective team- and cultural-assessment tool built specifically for use within the team context is Team Science™ ([www.myai.org](http://www.myai.org)). This tool has allowed me to be far more successful as a coach and consultant to my clients in that it quickly allows me to assess

the cultural dynamics, deficiencies, and opportunities in each team I work with.

Team Science™ ([Figure 17.1](#)) helps businesses and teams by:

- Optimizing each employee and revealing or her potential and strengths.
- Enabling managers and executives to build the right teams for the right projects.
- Allowing hiring managers to recruit and employ the right candidates based on cultural and strategic fit.
- Increasing awareness of team dynamics and collaboration styles.
- Helping with decision making and conflict resolution.
- Focusing on how to choose the right Leaders.
- Encouraging you to empower your employees.

**Figure 17.1** A Team's Cultural Plot and Dynamics using the Team Science™ Platform



## Example Case

I have utilized the Team Science™ solution with many of my clients. One of my most memorable occasions was helping a medium-sized company fundamentally change their organization. After having the entire organization take the assessment, we quickly understood why there were not only problems in their delivery of products on time, but also the apparent gaps that created miscommunication between teams.

We did five things with this company in the span of 1.5 months:

- 1.** We optimized each team and moved employees around to functional roles they were best suited for (i.e., moving a business analyst with a knack for detail into a quality-engineer role).
- 2.** We assigned employees new responsibilities we knew they would thrive and be successful at.
- 3.** We hired three new employees based on cultural fit to the company and not on their resume alone.
- 4.** We increased productivity in every single team by allowing each team member to fully understand how to best engage with their fellow team members.
- 5.** We improved almost all employees' work-life balance and job satisfaction (survey of job satisfaction sent out two months after employment of Team Science™).

Team Science™ has been the best instrument in my tool belt as a coach and consultant. It has allowed me to engage correctly with my clients, optimize their human capital, and empower and enable the right leaders to lead their company to high-performance. A major win!

# Chapter 18

## Scrum of Scrums

I love daily stand-ups. I love quick and efficient meetings that enable everyone to get on the same page and understand the work that lies ahead for the day. I love being able to communicate the team's daily goal and the highest-priority items that need to receive immediate attention. Most of this happens in the Daily Scrum, but there is some information that needs to be disseminated to managers or other teams. If you have multiple teams, cross-team issues, or other dependencies that need to be discussed, then it may be a cue to have a so-called Scrum of Scrums.

A Scrum of Scrums is a Scrum team made up of representatives from each of several other teams. In multiple code base teams, this is absolutely essential. I've consulted with IT departments that have multiple teams utilizing different code bases and have different managers for each team. How is it possible to keep up with all of the information shared between those teams, and the goals each team works toward, with their interdependencies and complex interplay of resources? Again, you need a Scrum of Scrums.

A Scrum of Scrums meeting can happen right after a regular Scrum meeting. Like any regular Scrum team, the Scrum of Scrums works iteratively to deliver value in the form of removing organizational and cross-functional team obstacles, dependencies, and conflicts. Just like a regular Scrum stand-up, the purpose of the Scrum of Scrums meeting is to resolve escalated obstacles or resolve dependencies between teams, as well as to document and follow up on any necessary communication between managers and developer leads on each of the respective teams.

### Example Case

I found this quick stand-up to be of great value in my experience with one client. It enabled me to communicate to all the lead developers and managers the overall goals and enterprise-wide needs, as well as to document and note issues the teams needed to resolve. This daily check-

in often provides valuable information to me because, remember, my primary job as an Agile coach is to remove obstacles to success! The interplay between cross-functional teams at times can be as complex as it is cumbersome. A meeting of the minds once each day can open up the flow of communication needed to break down the barriers that inhibit absolute productivity.

