CHAPTER ONE

1 What Is Agile?

AGILE IS A FRAMEWORK

It felt like a no-brainer to answer this question, as we set our sights on publishing a book on Agile auditing. Through discovery, we found that Agile has different meanings depending on your view and approach. When you develop and work with Agile, it's vital that you describe what Agile is and what it means.

Authors of other Agile publications describe it as a mindset or a methodology. Agile, for example, in Rick Wright's *Agile Auditing: Transforming the Internal Audit Process* (Wright 2019), he uses big "A" and little "a" to distinguish between *doing* Agile and *being* agile. Used as a noun, Wright refers to the big "A" as doing Agile internal auditing using software development methodologies. Wright's little "a," used as a verb, describes, in general, process improvement efforts (exclusive of specific methodology) to achieve a nimbler, less wasteful process. Big "A" is essentially the technical aspect of completing an audit. Little "a" is the thinking behind being agile. Being agile is as unique to an organization as your DNA is to you. To *do* Agile well, you must *be* agile, so from here on we make no distinction between being Agile and doing Agile. Agile is both a mindset and a framework. We hope that your organization, including your audit team, will demonstrate business agility using Agile methods. Agile organizations identify changes and risks from internal and external sources, respond to those changes promptly and appropriately,

deliver value to their customers, and remain sustainable. While this book is a framework providing options to implement Agile auditing, we've also provided various "recipes" with step-by-step examples of how to implement the framework. These recipes are as close as we get to prescribing a methodology. Remember, the recipes and the case studies provided in the text are just examples!

> *Agile* is not a methodology itself in any discipline. It is a philosophy, a mindset, or a way of thinking to get stuff done faster based on the interests of identified customers.

It is important to note that *Agile* is not a methodology itself in any discipline. It is a philosophy, a mindset, or a way of thinking to get stuff done faster based on the interests of identified customers. The roots of Agile as a philosophy originated in software development. It was software developers who combined existing frameworks to create the Agile movement to complete software development projects faster. You can think of Agile as an umbrella term for a set of different frameworks and practices all based on the original software development values and principles. These values are expressed in the "Manifesto for Agile Software Development," and the 12 principles as fashioned by the Agile Alliance are presented later in this chapter. Another key thought is that Agile methods are people-oriented rather than process-oriented. In Agile, people come first and people complete projects. Conversely, conventional project management and software development methods, such as waterfall, are process-oriented.

Before we continue describing Agile, we want to clarify that there is a time and place for traditional conventional project management methods, such as waterfall. For example, certain mandatory compliance audits with repeated processes year after year might benefit from a waterfall process-oriented approach. As a matter of fact, although there appears to be a mass adoption of various Agile methodologies in many organizations, there are still many that continue to use conventional methods successfully. We have also seen organizations transition into a hybrid Agile approach that combines aspects of both Agile and waterfall. Our Agile framework was developed specifically to help address common problems that arise when completing all audits using the traditional methodologies (i.e., waterfall).

DEFINITIONS OF AGILE

Agile is an approach to project management based on a set of values and principles. [The Agile approach] breaks projects into smaller, incremental deliverables that go through repeated iterations to focus on customers' needs and interests. It promotes adaptive planning, early delivery, frequent inspections, continuous improvement, and flexibility to respond to change (Catlin 2020).

Agile means quick, easy, and nimble. In business, it's a way of thinking, a way of working that is increasingly part of how many of the most successful companies work (Cazaly 2017).

Agile is the ability to move quickly and easily in response to your environment. To be Agile, you must be alert to your situations, and you must be flexible, nimble, and adaptable (Catlin 2014).

Agile is a lightweight software development method that aims to be more efficient than traditional, plan-driven development models. Agile seeks to do more with less:

- More team-level decision-making
- Faster development time
- Faster response to shifting customer demands
- Faster problem solving
- More customer satisfaction
- Smaller teams
- Less expense
- Less wasted effort
- Fewer features in the end product that either don't work or are never used (Mathis 2013)

As these definitions show, Agile is more than a project management tool, more than a mindset, more than an ability to be nimble and responsive, and more than a method to get things done. Table 1.1 contrasts three different views of Agile (LeMay 2018).

To understand Agile, it is helpful to examine the project life cycle and development life cycle. *A Guide to the Project Management Body of Knowledge* (*PMBOK® Guide* – Sixth Edition; Project Management Institute 2017) states that "Project life cycles can be predictive or adaptive. Within a project life cycle, there are generally one or more phases that are associated with the development of the product, service or result. These are called a development life cycle . . . and can be predictive, iterative, incremental, adaptive, or hybrid models."

TABLE 1.1 Views of Agile

Agile as a Methodology	Agile as a Mindset	Agile as a Movement
Practices matter more than mindset.	Mindset matters more than practices.	Mindset and practices are inexorably connected.
The practices and methods of Agile were already determined by others.	The principles and values of Agile were already determined by others.	I have an active role to play in determining how Agile principles and practices are articulated and applied in my team or organization.
Individuals within teams must collaborate and interact in prescribed and predefined ways.	Individuals within teams must independently develop an Agile mindset.	Individuals within team must work together toward a shared set of goals and values.

The *PMBOK® Guide* goes on to explain that:

- The traditional model, otherwise called waterfall, is a predictive life cycle where the project scope, time, and cost are determined in the early phases of the cycle. Any changes to scope are carefully managed. However, Agile is an adaptive life cycle and it may be iterative and incremental.
- "In an iterative life cycle, the project scope is generally determined early in the project life cycle, but time and cost estimates are routinely modified as the project team's understanding of the product increases. Iterations develop the product through a series of repeated cycles, while increments successively add to the functionality of the product." Iterative means repetitious actions. In Agile, iterations of project designing, planning, executing, testing, inspecting, and improving are repeated until a project is complete. The iterations increase functionality during a project until it is complete. In comparison, the waterfall includes single stages for designing (requirements definition), planning (project planning), testing/execution, and inspecting. In waterfall, the project constraints of scope, time, and cost are estimated at the beginning of a project. Agile frameworks determine and allow for flexibility on scope, time, and cost during an iteration.
- "In an incremental life cycle, the deliverable is produced through a series of iterations that successively add functionality within a predetermined time frame. The deliverable contains the necessary and sufficient capability to be considered complete only after the final iteration.

- "Adaptive life cycles are Agile, iterative, or incremental. The detailed scope is defined and approved before the start of an iteration. Adaptive life cycles are also referred to as Agile or change-driven life cycles.
- "A hybrid life cycle is a combination of a predictive and an adaptive life cycle. Those elements of the project that are well known or have fixed requirements follow predictive development life cycle, and those elements that are still evolving follow an adaptive development life cycle."

Project managers must select the right approach, be it traditional, iterative, incremental, adaptive, or hybrid, to manage the project throughout the project's life cycle. The project manager, or the project team in some approaches, must balance the constraints of time, scope/quality, and cost. Constraints are anything that restricts the actions of the project team (Heldman 2005). In projects using some of the Agile frameworks, we restrict the time (i.e., Agile projects should be less than one month), and often restrict the cost, and estimate the scope. The scope is what will be covered or achieved, as agreed with the customer, in a particular project. The scope includes the functions, features, data, content, standards, and so forth that will be provided during the project. In some Agile frameworks, scope is adjusted frequently to fit in the fixed time and cost constraints and teams do not add more time to complete Agile projects. In Agile, all constraints are value-driven, meaning one starts with the value proposition for Agile projects. Therefore, Agile projects focus only on necessary and enough processes to create the product. Moreover, Agile encourages adapting the project as needed to accomplish desired and changing goals.

Agile is also a mindset reflected in your organization's culture. To do Agile project management, you must begin with the Agile mindset (see Chapter 6, Creating the Agile Mindset). Your culture must support your efforts to execute your projects in an Agile way. We've dedicated an entire chapter to creating the Agile culture (see Chapter 17, Preparing Your Organization for Agile Auditing/Creating the Agile Culture) because the right culture is essential to the success of being and doing Agile. As Denning once stated, the wrong culture can kill any Agile efforts (Denning 2018). Use the following guiding principles as a foundation for the Agile mindset, methodology, and movement:

- Agile means we start with our internal or external customers and continuously consider customer needs. Customer needs change. Sometimes customers don't know what they need until they see what they do not need. An Agile team's role in completing projects is just as crucial as

determining customer needs and figuring out how to deliver value. Agile is customer-centric.

- Agile means we collaborate early and often. We start by identifying customer needs, write our requirements from the customer's view, and determine the work we can complete as a collaborative team. There is constant communication between team members. We do not wait until a specific meeting or time to communicate and collaborate. If you need to work with a Team Member, you let them know immediately. If you need something from a Team Member to complete your work, you tell them as soon as you know you need it, and preferably before you need it to complete your next task. Agile requires collaboration early in every project and frequently throughout the life of the project.
- Agile means we plan for uncertainty. We not only recognize uncertainty exists; we expect it and plan for it. We are equipped with the tools and knowledge needed to address ambiguity in processes and in customer needs. We are prepared for the unexpected because we planned for it. Agile is being prepared for change.

THE AGILE MANIFESTO

Let's continue our understanding of Agile with a little history lesson. Agile was created for software development projects by 17 software practitioners. Collectively, these practitioners are referred to as the Agile Alliance. These 17 individuals met in Snowbird, Utah, in February 2001 to "uncover better ways of developing software." The 17 software practitioners created the Agile Manifesto, which states (Beck et al. 2001):

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
> **Working software** over comprehensive documentation
> **Customer collaboration** over contract negotiation
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.

The Agile Alliance, officially formed in late 2001, is a nonprofit organization that promotes, disseminates, and develops the use of Agile and

supports people using Agile (Agile Alliance 2001). While the roots of Agile are in developing software, other industries, disciplines, functions, and professions use Agile, including marketing, sales, construction, event planning, and now auditing.

The four Agile Manifesto values set the basis for 12 principles as follows (Agile Alliance 2001):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-managing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The values and principles discussed earlier in this chapter are essential ingredients of the Agile mindset and set the foundation of Agile frameworks.

AGILE FRAMEWORKS

Many Agile enthusiasts consider Agile to be an umbrella for other types of rapid development and delivery options that foster collaboration among

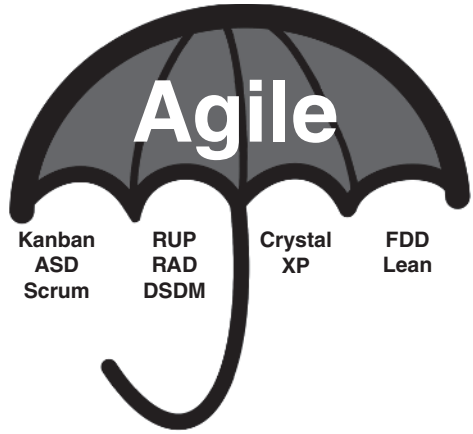

FIGURE 1.1 Agile Umbrella
Source: Illustration by Carmen Catlin.

cross-functional teams as depicted in Figure 1.1. Agile software development advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change (Agile Alliance 2001). When Agile was born in 2001, other methods to achieve faster development were already in existence, including the following (note, parenthesis in each bullet indicates the creation year and the creator's last name):

- Scrum (1986, Takeuchi and Nonaka [concept]; 1995, Schwaber and Sutherland)
- Extreme Programming (XP) (1996, Beck)
- Kanban (1953, Toyota)
- Rational Unified Process (RUP) (1987, Jacobson [concept]; 1996, Kruchten/IBM)
- Crystal (1996, Cockburn)
- Dynamic Systems Development Methodology (1995, DSDM Consortium)
- Feature Driven Development (FDD) (1997, De Luca)
- Rapid application development (1991, Martin)
- Adaptive software development (1974, Edmonds [concept]; 1995, Highsmith and Bayer)

After the 2001 Agile Manifesto, Lean software development was introduced in 2003 by Harry and Tom Poppendieck. We discuss Lean auditing

principles in Chapter 13, Lean and Agile Auditing. There are several Agile certifications available to demonstrate your competency with the various frameworks. While each of these methods is distinct, they all require constant communication and team collaboration. The following is a brief, high-level description of some of these methods:

- Scrum is a timebound Agile method where each task or activity is limited by a specified duration. Scrum is the most popular framework; we discuss it in more detail later in this chapter.
- XP focuses on developing high-quality software in a short period of time based on customer participation, rapid feedback, and subsequent planning and testing. It leverages four values: courage, simplicity, feedback, and communication.
- Kanban focuses on productivity and flow when there are no "features" released to a customer. It works well for small teams and is typically used in manufacturing processes. Kanban is not time-sensitive.
- Crystal is a collection of Agile methodologies and recognizes that any project may need a unique set of practices. It focuses on early and regular product delivery by increasing user participation and eliminating bureaucracy, while reinforcing the need for communication, teamwork, and simplicity.
- DSDM emphasizes business needs, active user participants, team empowerment, and constant delivery. Requirements are determined early in the project and refined as the project progresses.

In many projects, including those using Agile methods, effective communication often determines the success of the project. Each Agile method reinforces the importance of communication and most include more frequent meetings, such as daily meetings for Agile teams to coordinate activities. The daily meeting is a unique element of Agile project management for delivering results faster, correcting errors sooner, inspecting the project often, and overcoming obstacles before the obstacles become bottlenecks.

Raven's first introduction to the daily meeting was during a financial restatement project led by the chief administrative officer (CAO). To make sure participants didn't "get comfortable and talk too much," the CAO literally removed all the chairs from the room. Each Team Member simply and quickly stated the completed activities from the previous day, the planned activities for that day, and identified any obstacles preventing moving forward with the project. We never skipped a daily meeting, someone was always

present to discuss the work performed for each team, and it exceeded 15 minutes on only one occasion. In retrospect, the meeting was a Scrum of Scrums. A Scrum of Scrums is an Agile technique that integrates the work of multiple Scrum Teams. When there are many individuals in the team they are divided into smaller groups (usually five to nine members each), working on the same project. It allows the teams to communicate with each other to ensure that they are all working to accomplish the **Product Goal**. The output of each team is integrated with the output of all teams. This is vital in areas where there could be overlap or the sequencing of events is important. Ceciliana's first attempt to introduce the daily meetings and remove the chairs from the conference room during a project she was leading resulted in the participants becoming upset because they were going to have to limit the time they could speak and standing was not their preferred posture. After explaining that the reason for removing the chairs was to ensure a focused and speedy meeting, the calm compromised, and the chairs were returned on a condition of brevity. This is a great "lessons learned" example regarding the importance of clear and timely communications. In this case, even after allowing chairs in the meeting, meeting time was reduced from approximately two hours to 30 minutes, on average, and communications were limited to actions needed to complete the project. In both examples, the daily meeting was essential in timely project completion and is thus one technique you can use today to be Agile.

There was one specific meeting that solidified the daily meeting's value for Raven, which is the main reason we adopted the daily meeting in our Agile framework. Something was said during one of the meetings that could have changed the successful outcome of the restatement project and would have undoubtedly led to bottlenecks to complete the restatement effort. A team leader presented the accounting manager's obstacle – the need to approve thousands of manual journal entries in a very short time frame. Another team lead suggested that the approval was "easy, just select all and click approve." Talk about correcting project errors sooner! Needless to say, the CAO took immediate notice of the obstacle and the potential risk and implemented corrective actions and safeguards against such behaviors and mindsets. One corrective action included adding additional journal entry approvers. As a result, accounting managers ceased the "select all and click approve" approach to approve manual journal entries. Additionally, the auditors monitored to ensure that this behavior did not recur by using the accounting system's embedded audit module. The auditors created a rule to alert them if one accounting manager, or any person with approval privileges, approved a certain number of manual journal entries within a limited time frame.

If it weren't for the daily meeting to identify obstacles, there is a high probability that the two team leaders may have had a hallway conversation about the obstacle that week and the solution would have been "select all and click approve." To a control fanatic, this is obviously the wrong solution. We found so much value in the daily meeting on that project that we started using it in other projects, not realizing at the time that it was an Agile technique.

SCRUM FRAMEWORK

Scrum is the most popular Agile framework. Scrum is actually a rugby term. It is not an acronym. Since we've never personally played rugby, please bear with us on the following layman's description of a Scrum. In a rugby Scrum, when the ball is put into play, the referee drops the ball in the middle of two teams whose arms are interlocked together. In the huddle, as each member is bound to another, the team communicates instructions so the team can work in unison to get possession of the ball quickly. Each team has the same goal: to quickly get possession of the ball and move the ball to the goal. To achieve this, the team passes the ball back to a Team Member at the back of the Scrum using only their feet, referred to as a hook, as their arms remain bound to others. The team must work together quickly and efficiently to move the ball to the back of their team before the other team takes the ball. Then, the team with possession tries to score as the team moves collectively downfield as they pass the ball back and forth numerous times as needed to take advantage of player expertise, skill, and position to score points. This collective, quick, and efficient movement is similar to an Agile Scrum Team working together quickly and efficiently to achieve a specific goal.

> A key concept under Scrum is that a project's time and cost are set early in the project and scope is modified as new information and client needs are determined.

A key concept under Scrum is that a project's time and cost are set early in the project and scope is modified as new information and client needs are determined. Applying this concept to the rugby Scrum, each game (or project) lasts 80 minutes – the time – and each half (or increment) is limited to

40 minutes – the time. [As a side note, there are even some who believe that the Scrum time itself should be limited, as long Scrums are a "waste of time and increase chances of player injuries." Luckily in Scrum project management, the Scrum time is already limited to one to four weeks. One could say that by limiting the Scrum time in Agile/Scrum project management, we are reducing waste and risk to the organization.] The number of players on the field – the cost – is limited. The scope, or how many points the rugby team will score, and how they will actually maneuver to score points/goals are fluid during the game. Depending on the other team and the environment – this is usually the customer's role in a Scrum – the team adapts its scope and method during play. Additionally, depending on the strengths, weaknesses, and availabilities of the team members, the project scope and methods evolve throughout the game. Usually, the team that adapts the best, wins!

Ken Schwaber and Jeff Sutherland, co-creators of *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*, updated the guide in November 2020. According to the co-creators, "The 2020 version aimed to bring Scrum back to being a minimally sufficient framework by removing or softening prescriptive language" (Schwaber and Sutherland 2020). While developing our Agile auditing framework and writing this book, we refer to the 2017 version. However, although we have endeavored to incorporate the changes as reflected in the November 2020 update, you will notice references to both versions. This has no substantive effect on our framework; the essence of Scrum has not changed; Scrum is still Scrum. The latest update nonetheless makes Scrum more adaptable to all disciplines, not only software development, and continues to become more streamlined, lighter, and easier to understand. It is one team approach working together towards one Product Goal. As Jeff Sutherland says, "Scrum works best when it is fast, easy, and fun." Following is a summary of the differences between the 2017 and 2020 versions (Scrum Guides.org 2020):

- The 2020 version aimed to bring Scrum back to being a minimally sufficient framework by removing or softening prescriptive language.
- One Team, focused on one Product. There is just one Scrum Team focused on the same objective, with three different roles or sets of accountabilities: Project Owner, Scrum Master, and Developers (i.e., individuals performing the audit work). You will note that in this book we use the word "Roles" to denote the different team functions. However, within the Scrum Team, there are no subteams or hierarchies. It is a cohesive unit of professionals

focused on one objective at a time, the Product Goal. The 2020 updated Scrum Guide no longer refers to roles but rather to "Team."

- A Product Goal. The 2020 Scrum Guide introduces the concept of a Product Goal to provide focus for the Scrum Team toward a larger valuable objective. A Product Goal provides focus for the Scrum Team toward a larger valuable objective. Each Sprint should bring the product closer to the overall Product Goal.
- Sprint Goal, Definition of Done, and Product Goal. Previous Scrum Guides described Sprint Goal and Definition of Done without really giving them an identity. They were not quite Artifacts but were somewhat attached to Artifacts. With the addition of Product Goal, the 2020 version provides more clarity around this. Each of the three Artifacts now contain "commitments" to them. For the Product Backlog the commitment is the Product Goal, the Sprint Backlog commitment is the Sprint Goal, and the Increment commitment is the Definition of Done. They exist to bring transparency and focus toward the progress of each artifact.
- Self-managing over self-organizing. The 2020 version emphasizes a self-managing Scrum Team, choosing who, how, and what to work on.
- Three Sprint Planning Topics. In addition to the Sprint Planning topics of "What" and "How," the 2020 Scrum Guide places emphasis on a third topic, "Why," referring to the Sprint Goal.
- Overall simplification of language for a wider audience. The 2020 Scrum Guide has placed an emphasis on eliminating redundant and complex statements as well as removing any remaining inference to IT work (e.g., testing, system, design, requirement, etc.).

For the most part, the concepts listed in the summary of differences between the 2020 and 2017 Scrum Guide versions were already embedded in our Agile auditing framework. When they were not, efforts have been made to include the most current updates. We appreciate these changes as they simplify the job of implementing an Agile auditing framework, especially focusing on one goal and the removal of inferences to IT work. As auditors we can and will continue to strive for better communications and greater efficiencies eliminating redundant and complex language and minimizing the use of audit jargon in all of our communications.

Ken Schwaber and Jeff Sutherland identify three pillars of the Scrum framework that facilitate collaborative work environments (Schwaber and Sutherland 2020):

- *Transparency*. When a Team Member says something is done, it is really done. Finished. No loose ends. Transparency enables inspection. Inspection without transparency is misleading and wasteful.
- *Inspection*. A check on progress. An opportunity to identify problems and potential problems. Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.
- *Adaptation*. Implementing changes based on inspection. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

Additionally, there are five Scrum values (Scrum Alliance 2015). "These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. . . . When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust" (Schwaber and Sutherland 2020).

1. Commitment
2. Courage
3. Focus
4. Openness
5. Respect

> The three pillars of the Scrum framework (transparency, inspection, adaptation) facilitate collaborative work environments and the five values (commitment, courage, focus, openness, respect) give direction to the Scrum Team with regard to their work, actions, and behavior. When these five values are embodied by the Scrum Team and the people they work with, the Scrum pillars of transparency, inspection, and adaptation come to life building trust.
>
> (Schwaber and Sutherland 2020)

Each Team Member must demonstrate these pillars and values in the execution of their roles during the project.

Scrum/Agile Roles

In Scrum project management, there are only three recognized roles: one Scrum Master, one Product Owner, and various Developers. (The 2020 Scrum Guide explains that "Developer" does not mean software developers exclusively; it is intended to be an inclusive term. It refers to the members of the Scrum Team who are doing the work or developing the product. In this book we refer to a "Developer" as a "Delivery Team Member," which we have found more acceptable and more in alignment with audit terminology.) Collectively, those serving in these three roles create the Agile team (the equivalent to a Scrum Team). The Agile team members are self-organizing, to ensure the best complement of skills, knowledge, and capabilities, and self-managing, to increase accountability and workability of the team without the need for a manager, or, worse, a micromanager. There are no subteams or hierarchies. The team is a cohesive unit of professionals focused on one objective at a time, the Product Goal. The Agile team has total authority on the exact approach to get their work done, estimate how long work will take, create their schedule, and manage their own time. The Agile team is small enough to remain nimble and large enough to complete significant work within a Sprint. Most Agile teams have three to nine members. The Agile team is accountable for all aspects of the work (Rigby, Sutherland, and Takeuchi 2016). Following is a brief description of each role.

Product Owner

The **Product Owner** owns the "what" of the project. They create a prioritized list of all the things to need to get done (**Product Backlog**). The Product Owner maximizes the value of the product being delivered (the audit). The Product Owner determines the priorities for Agile teams and decides when a product is complete by assessing value from the customer's perspective. In some Agile variants, the Product Owner is referred to as the "initiative owner" (Rigby, Sutherland, and Takeuchi 2016). The Product Owner is not typically the customer, but someone who represents the customer's interest, voice, and mindset during the project. The Product Owner *must* have the Agile mindset; this is a critical role. Agile Teams cannot select just any manager to serve as the Product Owner in the same way managers are currently selected to manage traditional teams and projects. Even if one is a highly competent manager, that does not mean they have the right mindset and skills to be a successful Product Owner. Choosing the right Product Owner is as important as selecting an Agile framework. In the words of J.J. Sutherland, "if you keep doing things the way you

have always done you will get the results you always got before. Good Product Owners are the key to winning with Scrum" (Sutherland 2019).

The Product Owner must understand the customer and the customer's needs. The Product Owner owns the Product Backlog (also known as a "portfolio backlog"), which is a list of requirements and deliverables for a project. The Product Owner works directly with the business community, stakeholders, customers, and users to obtain an understanding of the community and user needs. "For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review" (Schwaber and Sutherland 2020). The Product Owner prioritizes the Product Backlog list (see Chapter 8, Implementing Agile Auditing: The Audit Planning Process for more details on Product Backlogs).

Scrum Master

The **Scrum Master** owns the process and is held accountable for enhancing team performance. The 2020 update to the Scrum Guide elevates the role of the Scrum Master from a servant leader to that of a true leader who serves the Scrum Team and the larger organization (Schwaber and Sutherland 2020). In the 2020 Guide, the switching around of the words from a "servant leader" to "a leader who serves" is intended to recognize that the Scrum Master helps focus on the leadership role of the Scrum Master to help achieve the project goals. The Scrum Master is accountable for the Scrum Team's effectiveness by enabling the Scrum Team to improve its practices, within the Scrum framework. Nonetheless, the Scrum Master is not a boss, a project manager, or a decision-maker. The Scrum Master is a coach, facilitator, and an Agile champion. The Scrum Master is accountable for the Agile team happiness and why this is important. They remind the Agile team members of the benefits and value of the Agile approach whenever needed and encourage the right behaviors from the team members. For example, the Scrum Master may remind team members to be honest and transparent in their communications. The Scrum Master facilitates all meetings, including daily meetings. They coach the Agile team members in self-management and cross-functionality; help the team focus on creating high-value increments that meet the Definition of Done; and facilitate the removal of impediments or roadblocks and obstacles to help the team progress and remain efficient in completing Sprints, ensuring that all Scrum events take place and are positive, productive, and kept within the timebox. A **Sprint** is a short, timebound cycle within which team members complete an

increment of work to deliver products the customer needs based on whatever is most important to the customer at the time.

Delivery Team (Developers)

The **Delivery Team** Members own the "how." They complete product tasks during Sprints. They are the individuals in the Agile team who are committed to creating any aspect of a usable Increment in each Sprint. They create a **Sprint Backlog**, a specific, focused list of tasks determined by the Delivery Team to complete an increment that guides their Sprint/increment work. An *increment* is a product deliverable, usually a small portion of the overall product. The Sprint Backlog is a specific, focused list of tasks determined by the Delivery Team Members to complete an increment. The team has total authority on the exact approach to get their work done, estimate how long work will take, create their schedule, and manage their own time. They are a cross-functional, self-managing group of autonomous individuals collectively possessing all the skills necessary to complete the Sprint backlog. In Scrum, Developers can be business developers, user experience researchers, customer experience specialists, mechanical engineers, lab technicians, doctors, nurses, carpenters, marketers, researchers, scientists, quality assurance specialists, and more.

We will review these roles, responsibilities, and options throughout the book. In Chapter 7, Implementing Agile Auditing: Deciding Your Approach and Your Agile Audit Project Roles, we will address the different roles of Product Backlog in an auditing context. In Chapter 9, Implementing Agile Auditing: Planning Agile Audit Engagements, we will apply Scrum concepts in an audit context, including the Product Owner, Scrum Master, Delivery Team, documents, and activities. Remember, Agile and Scrum are frameworks, and adaptations are expected; some Scrum implementations include additional roles.

Scrum Artifacts

Scrum Artifacts represent work value. Following are the three Artifacts, or documents, created in the Scrum framework (Sutherland and Sutherland 2014). Each of the three Scrum Artifacts has a corresponding commitment to drive focus and alignment. This commitment gives the teams a much better focus on the specific goals. The Artifacts are:

1. **Product Backlog** (see Chapter 8, Implementing Agile Auditing: The Audit Planning Process). The Product Backlog is a list of requirements and

features for a project that is managed by the Product Owner in order of business priority. Product Backlogs include estimates on business value and development efforts. The commitment for Product Backlog is Product Goal.

2. **Sprint Backlog** (see Chapter 9, Implementing Agile Auditing: Planning Agile Audit Engagements). The Sprint Backlog is a specific, focused list of tasks the Delivery Team believes it can complete in a Sprint. It is created by the team members, using a *pull* approach to complete an increment. Contrasted with a *push* approach, where an input is pushed into a cycle in hopes that it can be used as it is pushed, or can wait until it is needed, a pull approach pulls inputs into the process or production line on demand, as needed. The push approach may result in excessive production and unused work, while the pull approach is quick and efficient. The commitment for Spring Backlog is Sprint Goal.
3. **Increment.** An increment is production output at the end of a timeboxed Sprint. The commitment for increment is the Definition of Done.

We acknowledge that some Scrum adaptations include up to six Artifacts, or documents. However, creating more Artifacts that do not add value or that are otherwise created simply for the sake of creating more Artifacts does not align with the Agile Manifesto value of "more working software, less documentation."

Scrum Activities (Scrum Events)

The ideas presented in this section align with the typical projects that use the Scrum framework. The Scrum framework organizes work into one- to four-week increments called Sprints. The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to examine and adapt Scrum Artifacts. These activities are specifically designed to enable the transparency required. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum.

Optimally, all events should be held at the same time and place to reduce complexity and increase effectiveness of the team. Each Sprint should bring the product closer to the overall Product Goal and likely will evolve as the Scrum Team learns over time. All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, Sprint Retrospective, and Product Backlog refinement/grooming, happen within Sprints. The Product Owner is the only one who can abort the Sprint. There is a specific time frame for each Sprint and a time frame to complete each of the

Sprint activities described in the list that follows. The Sprint duration determines the time frame, or "timebox," for each Sprint activity as noted in the description for each activity. We define and relate each of the five following Scrum Activities to audit activities in Part II:

1. **Sprint Planning Meeting** (Chapter 9). The Sprint planning meeting is a timeboxed activity (two hours or less per week of Sprint length) held at the beginning of the Sprint to determine the features to be delivered in each Sprint. It is facilitated by the Scrum Master. The Product Owner is an active participant who provides clarity on the upcoming project and related customer stories. The team members collaborate to determine the Sprint tasks, a Definition of *Done*, and a Definition of *Ready*. For a two-week Sprint, the Sprint planning meeting would occur over a four-hour timebox. For a four-week Sprint, the Sprint planning meeting would occur over an eight-hour timebox.
2. **Daily Meeting** (Chapter 11). The daily meeting may also be called a Daily Sprint, Daily Scrum, or daily standup. For a two-week Sprint, the daily meeting lasts no more than 15 minutes; for a four-week Sprint, the daily meeting lasts no more than 30 minutes. The Scrum Master facilitates the meeting, which is held virtually or in a public location at the same time and place each day. Team members take part, provide updates, and give feedback during the meeting. The meeting helps increase transparency on the Sprint and increases communication among the team members. While only the development team provides updates, others may observe the meeting. The Scrum Master ensures meeting productivity and limits unnecessary contributions, updates, and questions.
3. **Sprint Review** (Chapter 11). The Sprint Review is the Scrum Delivery Team's presentation of their increment, or product, that will be provided to the customer. The development team also provides a summary of the increment and any incomplete tasks. The Product Owner has the authority to approve the increment during the Sprint Review. This review meeting occurs at the end of a Sprint. The Sprint Review is timeboxed to one hour or less for every week of Sprint length. For a two-week Sprint, the Sprint Review is limited to two hours.
4. **Sprint Retrospective** (Chapter 11). The Sprint Retrospective is a "lessons learned" and continuous improvement meeting that lasts one-hour for a two-week Sprint. Team members, Scrum Masters, and the Product Owner discuss what worked well and what can be improved on the next Sprint. This one-hour meeting is held immediately following the Sprint Review.

5. **Product Backlog Refinement/Grooming** (Chapter 8). The Product Backlog refinement meeting is led by the Product Owner, who engages the team members, Scrum Master, stakeholders, and others to determine the next highest priority item(s) on the Product Backlog for the next Sprint. Typically, this two-hour meeting occurs the morning after a Sprint Retrospective.

Nevertheless, while the durations provided in this list represent typical projects that use the Scrum framework, our Agile audit framework does not subscribe to doubling the time frame in the timeboxed activities unless there is a legitimate and valuable reason to do so. Additionally, we consistently use a two-week Sprint.

Figure 1.2 visually depicts how the Scrum roles, Artifacts, and activities fit together. In Chapter 4, What Is Agile Audit?, we alter this diagram for the Agile auditing framework.

There is plenty of evidence supporting Agile in any form. According to an article by Consultancy.eu (a European online platform for the advisory and consulting industry), a study by Organize Agile among professionals in 19 countries reported that 83% of respondents said "it is the ability to improve flexibility amid a rapidly changing environment" that makes Agile appealing. "Leveraging a quicker way of bringing incremental innovations and new products/services to the market, companies can timely cater to the changing needs of customers and try to stay ahead of their competition. Sticking to the traditional Waterfall approach in today's environment often means that organizations are left a step behind of their competition" (Consultancy.eu 2020). Auditing must adopt Agile practices to keep up with business needs and avoid being outsourced, or worse, eliminated tomorrow.

Even the Project Management Institute's flagship publication and fundamental resource for effective project management in any industry, the PMBOK (currently in its sixth edition), has been updated to include information on Agile practices alongside traditional approaches with its guidance. It states, "So why an Agile Practice Guide and why now? Project teams have used Agile techniques and approaches in various forms for at least several decades. The Agile Manifesto expressed definitive values and principles of Agile as the use of Agile gained substantial momentum. Today, project leaders and teams find themselves in an environment disrupted by exponential advances in technology and demands from customers for more immediate delivery of value. Agile techniques and approaches effectively manage disruptive technologies. In addition, the first principle of Agile places

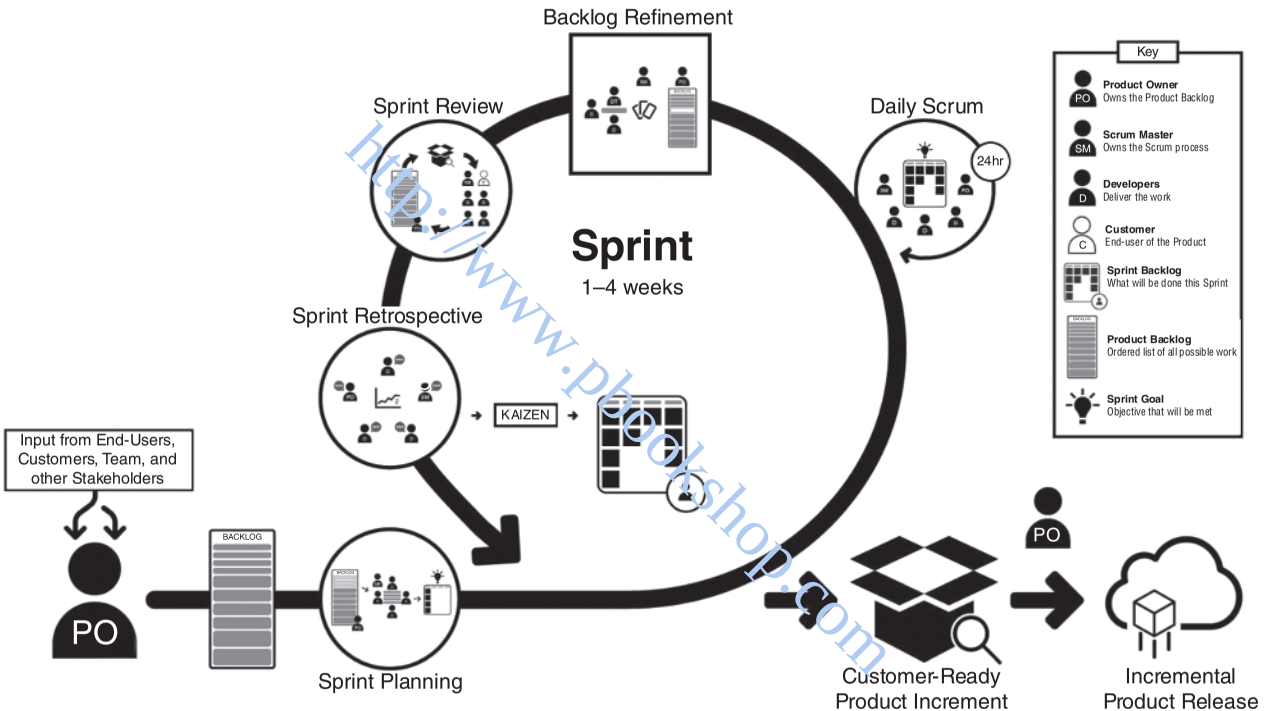

FIGURE 1.2 How Scrum Works

Source: "How Scrum Works: Scrum Framework," Scrum Inc., https://www.scruminc.com/scrum-framework/.

customer satisfaction as the highest priority and is key in delivering products and services that delight customers. Rapid and transparent customer feedback loops are readily available with the widespread use of social media. Therefore, in order to stay competitive and relevant, organizations can no longer be internally focused but rather need to focus outwardly to the customer experience" (Project Management Institute 2017).

RECIPE: EXPLAINING THE CONCEPT

We are both cooks, or home chefs if you prefer. Sometimes we are bakers, when we attempt to re-create some of our favorite desserts from shows like *Nailed It*, *Sugar Rush*, or *Cake Boss*. As chefs and bakers, we start with a recipe. Usually, the first time we make a dish, it is "by the book" and we strictly follow the recipe . . . unless we lack an ingredient in our pantry. If this is the case, we are Agile and substitute another ingredient or simply make do without it. After the first or second time cooking the dish, we start to look for variations to improve the dish. Sometimes the variations are suggestions from others. Sometimes they are based on personal experience. Sometimes they are driven by food allergies, availability of pantry items, or other necessities.

This approach to using a recipe is analogous to your approach for your Agile audit recipe. Sometimes the recipe is perfectly designed, sometimes the variations make the product better, and sometimes the variations are a complete and utter disaster. Your experiments and learning experiences will drive continuous improvement and each successive dish you prepare will get better and better. Experiment and learn with Agile. This will make you a better Agile auditor and enable you to create better dishes to serve your customers. This is comparable to Scrum; rather than providing detailed instructions, the rules of Scrum guide people's relationships and interactions.

NUGGETS

Agile is a project management approach to breaking down projects into smaller deliverables. It is a mindset of being able to move quickly in response to your environment. There are many frameworks and approaches to Agile project management, including Scrum, which is the most popular. The Agile Manifesto and Agile principles establish a foundation to build an Agile approach for your organization, regardless of which framework you select. Scrum includes

a set of values and attempts to create value by collaborating with the customer, streamlining projects, and building efficiency using three roles, three Artifacts, and five activities in every Scrum project. Scrum projects include Sprints. Each Sprint lasts from one week to four weeks. Self-managing teams complete each Sprint. The team creates a Sprint Backlog to clarify the Sprint tasks the team intends to complete during the Sprint. The Sprint Backlog includes a series of iterations to design, plan, test, inspect, and improve the next iteration and Sprint. Once the Sprint Backlog tasks are complete, the Sprint is over, and a new Sprint begins. The Sprint iterations repeat until the team completes the Scrum project and provides an increment, or product, to the customer. An important aspect to remember is that while implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices. As you begin your transformation, take a look at the recipes to help you determine your methodology. Don't be afraid to adjust the recipes!