1. INTRODUCTION

A Guide to the Scrum Body of Knowledge (SBOK™ Guide) provides guidelines for the successful implementation of Scrum—the most popular Agile project management and product development methodology. It provides a comprehensive framework that includes the principles, aspects, and processes of Scrum.

Scrum, as defined in the *SBOK™ Guide*, is applicable to the following:

- Portfolios, programs, and/or projects in *any* industry
- Products, services, or any other results to be delivered to stakeholders
- Projects of any size or complexity

The term “product” in the *SBOK™ Guide* may refer to a product, service, or other deliverable. Scrum can be applied effectively to any project in any industry—from small projects or teams with as few as six team members to large, complex projects with up to several hundred team members.

This first chapter describes the purpose and framework of the *SBOK™ Guide* and provides an introduction to the key concepts of Scrum. It contains a summary of Scrum principles, Scrum aspects and Scrum processes. Chapter 2 expands on the six Scrum principles which are the foundation on which the Scrum framework is based. Chapters 3 through 7 elaborate on the five Scrum aspects that must be addressed throughout any project: organization, business justification, quality, change, and risk. Chapters 8 through 12 cover the 19 Scrum processes involved in carrying out a Scrum project. These processes are part of the five Scrum phases: Initiate; Plan and Estimate; Implement, Review and Retrospect; and Release. These phases describe in detail the associated inputs and outputs of each process, as well as the various tools that may be used in each process. Some inputs, tools, and outputs are mandatory and are indicated as such; others are optional depending on the specific project, organizational requirements, and/or guidelines set forth by the organization’s Scrum Guidance Body (SGB). Finally, Appendix A contains an overview of *The Agile Manifesto* (Fowler and Highsmith, 2001) and a discussion of various Agile methods for those who want more information about Agile.

This chapter is divided into the following sections:

1.1 Overview of Scrum

1.2 Why Use Scrum?

1.3 Purpose of the *SBOK™ Guide*

1.4 Framework of the *SBOK™ Guide*

1.5 Scrum vs. Traditional Project Management

1.1 Overview of Scrum

A Scrum project involves a collaborative effort to create a new product, service, or other result as defined in the Project Vision Statement. Projects are impacted by constraints of time, cost, scope, quality, resources, organizational capabilities, and other limitations that make them difficult to plan, execute, manage, and ultimately succeed. However, successful implementation of the results of a finished project provides significant business benefits to an organization. It is therefore important for organizations to select and practice an appropriate project management methodology.

Scrum is one of the most popular Agile methodologies. It is an adaptive, iterative, fast, flexible, and effective methodology designed to deliver significant value quickly and throughout a project. Scrum ensures transparency in communication and creates an environment of collective accountability and continuous progress. The Scrum framework, as defined in the *SBOK™ Guide*, is structured in such a way that it supports product and service development in all types of industries and in any type of project, irrespective of its complexity.

A key strength of Scrum lies in its use of cross-functional, self-organized, and empowered teams who divide their work into short, concentrated work cycles called Sprints. Figure 1-1 provides an overview of a Scrum project's flow.

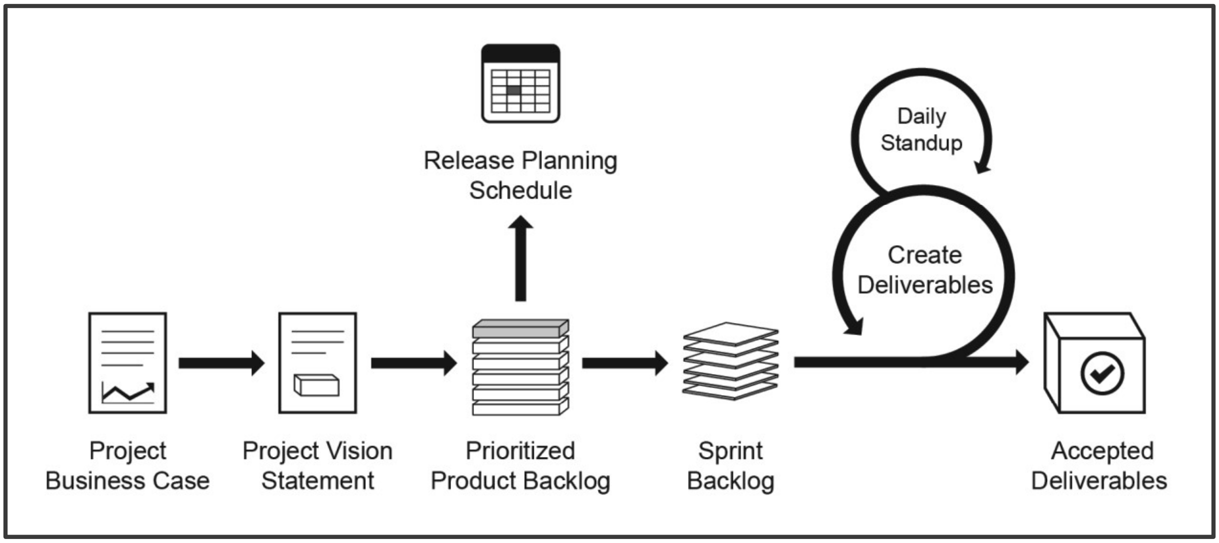

Figure 1-1: Scrum Flow for one Sprint

The Scrum cycle begins with a Stakeholder Meeting, during which the Project Vision is created. The Product Owner then develops a Prioritized Product Backlog which contains a prioritized list of business and project requirements written in the form of User Stories. Each Sprint begins with a Sprint Planning Meeting during which high priority User Stories are considered for inclusion in the Sprint. A Sprint generally lasts between one and six weeks and involves the Scrum Team working to create potentially shippable Deliverables or product increments. During the Sprint, short, highly focused Daily Standup Meetings are conducted where team members discuss daily progress. Toward the end of the Sprint, a Sprint Review Meeting is held during which the Product Owner and relevant stakeholders are provided a demonstration of the Deliverables. The Product Owner accepts the Deliverables only if they meet the predefined Acceptance Criteria. The Sprint cycle ends with a Retrospect Sprint Meeting where the team discusses ways to improve processes and performance as they move forward into the subsequent Sprint.

1.1.1 Brief History of Scrum

In the mid 80's, Hirotaka Takeuchi and Ikujiro Nonaka defined a flexible and all-inclusive product development strategy where the development team works as a unit to reach a common goal. They described an innovative approach to product development that they called a holistic or "rugby" approach, "where a team tries to go the distance as a unit, passing the ball back and forth." They based their approach on manufacturing case studies from various industries. Takeuchi and Nonaka proposed that product development should not be like a sequential relay race, but rather should be analogous to the game of rugby where the team works together, passing the ball back and forth as they move as a unit down the field. The rugby concept of a "Scrum" (where a group of players form together to restart the game) was introduced in this article to describe the authors' proposal that product development should involve "moving the Scrum downfield".

Ken Schwaber and Jeff Sutherland elaborated on the Scrum concept and its applicability to software development in a presentation at the Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) conference held in 1995 in Austin, Texas. Since then, several Scrum practitioners, experts, and authors have continued to refine the Scrum conceptualization and methodology. In recent years, Scrum has increased in popularity and is now the preferred project development methodology for many organizations globally.

1.2 Why Use Scrum?

Some of the key benefits of using Scrum in any project are:

1. **Adaptability—**Empirical process control and iterative delivery make projects adaptable and open to incorporating change.

2. **Transparency—**All information radiators like a Scrumboard and Sprint Burndown Chart are shared, leading to an open work environment.

3. **Continuous Feedback—**Continuous feedback is provided through the *Conduct Daily Standup,* and *Demonstrate and Validate Sprint* processes.

4. **Continuous Improvement—**The deliverables are improved progressively Sprint by Sprint, through the *Groom Prioritized Product Backlog* process.

5. **Continuous Delivery of Value**—Iterative processes enable the continuous delivery of value through the *Ship Deliverables* process as frequently as the customer requires.

6. **Sustainable Pace—**Scrum processes are designed such that the people involved can work at a sustainable pace that they can, in theory, continue indefinitely.

7. **Early Delivery of High Value—**The *Create Prioritized Product Backlog* process ensures that the highest value requirements of the customer are satisfied first.

8. **Efficient Development Process—**Time-boxing and minimizing non-essential work leads to higher efficiency levels.

9. **Motivation—**The *Conduct Daily Standup* and *Retrospect Sprint* processes lead to greater levels of motivation among employees.

10. **Faster Problem Resolution—**Collaboration and colocation of cross-functional teams lead to faster problem solving.

11. **Effective Deliverables—**The *Create Prioritized Product Backlog* process and regular reviews after creating deliverables ensures effective deliverables to the customer.

12. **Customer Centric—**Emphasis on business value and having a collaborative approach to stakeholders ensures a customer-oriented framework.

13. **High Trust Environment—***Conduct Daily Standup* and *Retrospect Sprint* processes promote transparency and collaboration, leading to a high trust work environment ensuring low friction among employees.

14. **Collective Ownership—**The *Approve, Estimate, and Commit User Stories* process allows team members to take ownership of the project and their work leading to better quality.

15. **High Velocity—**A collaborative framework enables highly skilled cross-functional teams to achieve their full potential and high velocity.

16. **Innovative Environment—**The *Retrospect Sprint* and *Retrospect Project* processes create an environment of introspection, learning, and adaptability leading to an innovative and creative work environment.

1.2.1 Scalability of Scrum

To be effective, Scrum Teams should ideally have six to ten members. This practice may be the reason for the misconception that the Scrum framework can only be used for small projects. However, it can easily be scaled for effective use in large projects. In situations where the Scrum Team size exceeds ten people, multiple Scrum Teams can be formed to work on the project. The *Convene Scrum of Scrums* process facilitates coordination among the Scrum Teams, enabling effective implementation in larger projects.

Large or complex projects are often implemented as part of a program or portfolio. The Scrum framework can also be applied to manage even programs and portfolios. The logical approach of the guidelines and principles in this framework can be used to manage projects of any size, spanning geographies and organizations. Large projects may have multiple Scrum Teams working in parallel making it necessary to synchronize and facilitate the flow of information and enhance communication. The *Convene Scrum of Scrums* is the process ensuring this synchronization. The various Scrum Teams are represented in this meeting and the objectives are to provide updates about progress, discuss challenges faced during the project, and coordinate activities. There are no set rules regarding the frequency of these meetings. The factors determining the frequency are the amount of inter-team dependency, size of the project, level of complexity, and recommendations from the Scrum Guidance Body.

1.3 Purpose of the *SBOK™ Guide*

In recent years, it has become evident that organizations which use Scrum as their preferred project delivery framework consistently deliver high Returns on Investment. Scrum's focus on value-driven delivery helps Scrum Teams deliver results as early in the project as possible.

The *SBOK™ Guide* was developed as a means to create a necessary guide for organizations and project management practitioners who want to implement Scrum, as well as those already doing so who want to make needed improvements to their processes. It is based on experience drawn from thousands of projects across a variety of organizations and industries. The contributions of many Scrum experts and project management practitioners have been considered in its development.

The *SBOK™ Guide* is especially valuable:

- to Scrum Core Team members including:
    - Product Owners who want to fully understand the Scrum framework and particularly the customer or stakeholder-related concerns involving business justification, quality, change, and risk aspects associated with Scrum projects.
    - Scrum Masters who want to learn their specific role in overseeing the application of Scrum framework to Scrum projects.
    - Scrum Team members who want to better understand Scrum processes and the associated tools that may be used to create the project's product or service.
- as a comprehensive guide for all Scrum practitioners working on Scrum projects in any organization or industry.
- as a reference source for anyone interacting with the Scrum Core Team, including but not limited to the Portfolio Product Owner, Portfolio Scrum Master, Program Product Owner, Program Scrum Master, Scrum Guidance Body, and Stakeholders (i.e., sponsor, customer, and users).
- as a handbook for any person who has no prior experience or knowledge of Scrum framework but wants to learn more about the subject.

The content of the *SBOK™ Guide* is also helpful for individuals preparing to write the following SCRUMstudy™ certification exams:

- Scrum Developer Certified (SDC™)
- Scrum Master Certified (SMC™)
- Agile Expert Certified (AEC™)
- Scrum Product Owner Certified (SPOC™)
- Expert Scrum Master (ESM™)

1.4 Framework of the *SBOK™ Guide*

The *SBOK™ Guide* is broadly divided into the following three areas:

1. **Principles** covered in chapter 2, expand on the six principles which form the foundation on which Scrum is based.

2. **Aspects** covered in chapters 3 through 7 describe the five aspects that are important considerations for all Scrum projects.

3. **Processes** covered in chapters 8 through 12 include the nineteen Scrum processes and their associated inputs, tools, and outputs.

Figure 1-2 illustrates the *SBOK™ Guide* framework, which shows that principles, aspects, and processes interact with each other and are equally important in getting a better understanding of the Scrum framework.

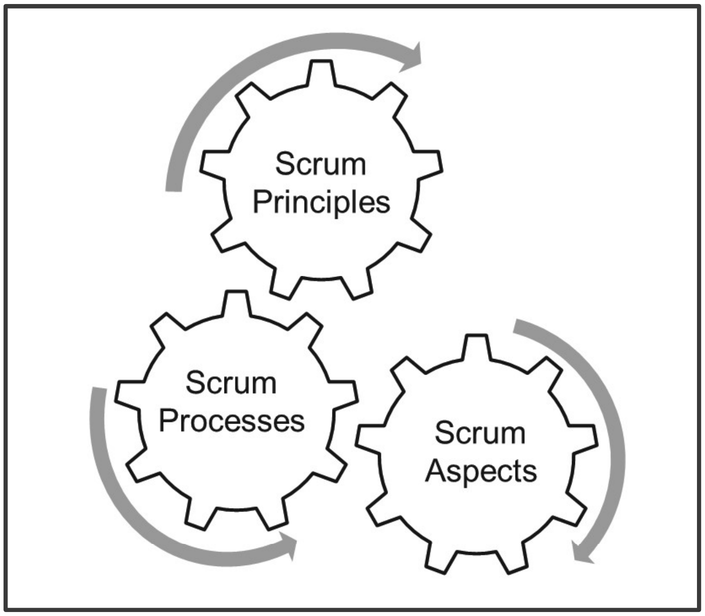

Figure 1-2: ***SBOK™ Guide*** **Framework**

1.4.1 How to Use the *SBOK™ Guide*?

The *SBOK™ Guide* can be used as a reference and knowledge guide by both experienced Scrum and other product and service development practitioners, as well as by persons with no prior experience or knowledge of Scrum or project management methodology. The contents are organized for easy reference by the three Scrum Core Team roles: Scrum Master, Product Owner, and Scrum Team.

The chapters covering the six Scrum principles (chapter 2) and five Scrum aspects (chapter 3 through 7) include a Roles Guide. This guide provides direction regarding the relevance of each section in the chapter to the Scrum Core Team roles.

In order to facilitate the best application of the Scrum framework, the *SBOK™ Guide* has clearly differentiated mandatory inputs, tools, and outputs, from non-mandatory or optional ones. Inputs, tools, and outputs denoted by asterisks (*) are mandatory while others with no asterisks are optional. It is recommended that those being introduced to Scrum focus primarily on the mandatory inputs, tools, and outputs, while more experienced practitioners should read the entire process chapters.

1.4.2 Scrum Principles

Scrum principles are the core guidelines for applying the Scrum framework and should mandatorily be used in all Scrum projects. The six Scrum principles presented in chapter 2 are:

1. Empirical Process Control
2. Self-organization
3. Collaboration
4. Value-based Prioritization
5. Time-boxing
6. Iterative Development

Figure 1-3 illustrates the six Scrum principles.

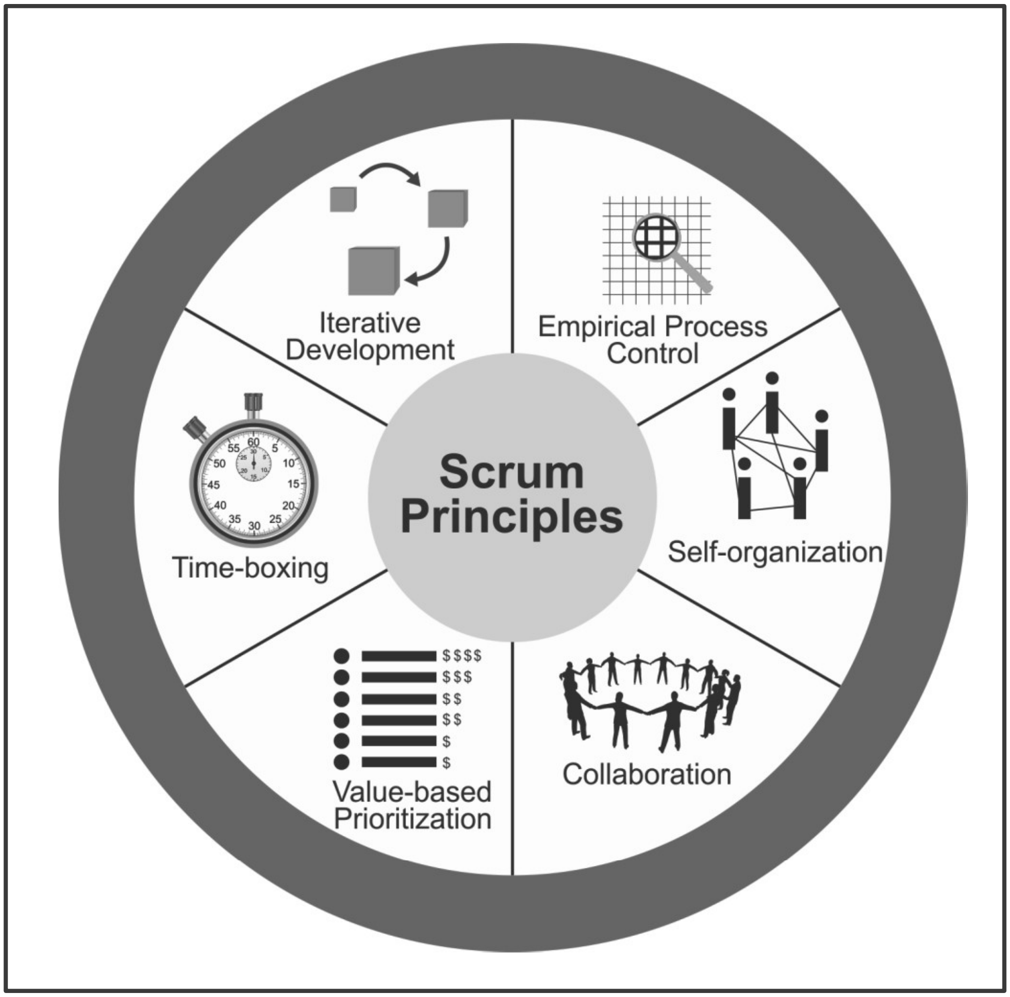

Figure 1-3: Scrum Principles

Scrum principles can be applied to any type of project in any organization and must be adhered to in order to ensure effective implementation of the Scrum framework. Scrum Principles are non-negotiable and must be applied as specified in the *SBOK™ Guide*. Keeping the principles intact and using them appropriately instills confidence in the Scrum framework with regard to attaining the objectives of the project. The Scrum aspects and processes, however, can be modified to meet the requirements of the project or the organization.

1. **Empirical Process Control**—This principle emphasizes the core philosophy of Scrum based on the three main ideas of transparency, inspection, and adaptation.

2. **Self-organization—**This principle focuses on today's workers, who deliver significantly greater value when self-organized and this results in better team buy-in and shared ownership; and an innovative and creative environment which is more conducive for growth.

3. **Collaboration—**This principle focuses on the three core dimensions related to collaborative work: awareness, articulation, and appropriation. It also advocates project management as a shared value-creation process with teams working and interacting together to deliver the greatest value.

4. **Value-based Prioritization—**This principle highlights the focus of Scrum to deliver maximum business value, from early in the project and continuing throughout.

5. **Time-boxing—**This principle describes how time is considered a limiting constraint in Scrum, and used to help effectively manage project planning and execution. Time-boxed elements in Scrum include Sprints, Daily Standup Meetings, Sprint Planning Meetings, and Sprint Review Meetings.

6. **Iterative Development—**This principle defines iterative development and emphasizes how to better manage changes and build products that satisfy customer needs. It also delineates the Product Owner's and organization's responsibilities related to iterative development.

1.4.3 Scrum Aspects

The Scrum aspects must be addressed and managed throughout a Scrum project. The five Scrum aspects presented in chapter 3 through 7 are:

1.4.3.1 Organization

Understanding defined roles and responsibilities in a Scrum project is very important for ensuring the successful implementation of Scrum.

Scrum roles fall into two broad categories:

1. **Core Roles—**Core roles are those roles which are mandatorily required for producing the project's product or service. Individuals who are assigned core roles are fully committed to the project and are ultimately responsible for the success of each project iteration and of the project as a whole.

   These roles include:

   - The **Product Owner** is the person responsible for achieving maximum business value for the project. He or she is also responsible for articulating customer requirements and maintaining business justification for the project. The Product Owner represents the Voice of the Customer.

   - The **Scrum Master** is a facilitator who ensures that the Scrum Team is provided with an environment conducive to complete the project successfully. The Scrum Master guides, facilitates, and teaches Scrum practices to everyone involved in the project; clears impediments for the team; and, ensures that Scrum processes are being followed.

   - The **Scrum Team** is the group or team of people who are responsible for understanding the requirements specified by the Product Owner and creating the Deliverables of the project.

2. **Non-core Roles—**Non-core roles are those roles which are not mandatorily required for the Scrum project and may include team members who are interested in the project. They have no formal role in the project team and may interface with the team, but may not be responsible for the success of the project. The non-core roles should be taken into account in any Scrum project.

   Non-core roles include the following:

   - **Stakeholder(s),** which is a collective term that includes customers, users, and sponsors, frequently interface with the Scrum Core Team, and influence the project throughout the project's development. Most importantly, it is for the stakeholders that the project produces the collaborative benefits.

   - **Scrum Guidance Body** (SGB) is an optional role, which generally consists of a set of documents and/or a group of experts who are typically involved with defining objectives related to quality, government regulations, security, and other key organizational parameters. This SGB guides the work carried out by the Product Owner, Scrum Master, and Scrum Team.

   - **Vendors**, including external individuals or organizations, provide products and/or services that are not within the core competencies of the project organization.

   - **Chief Product Owner** is a role in bigger projects with multiple Scrum Teams. This role is responsible for facilitating the work of multiple Product Owners, and maintaining business justification for the larger project.

   - **Chief Scrum Master** is responsible to coordinate Scrum-related activities in large projects which may require multiple Scrum Teams to work in parallel.

Figure 1-4 illustrates the Scrum Organization structure.

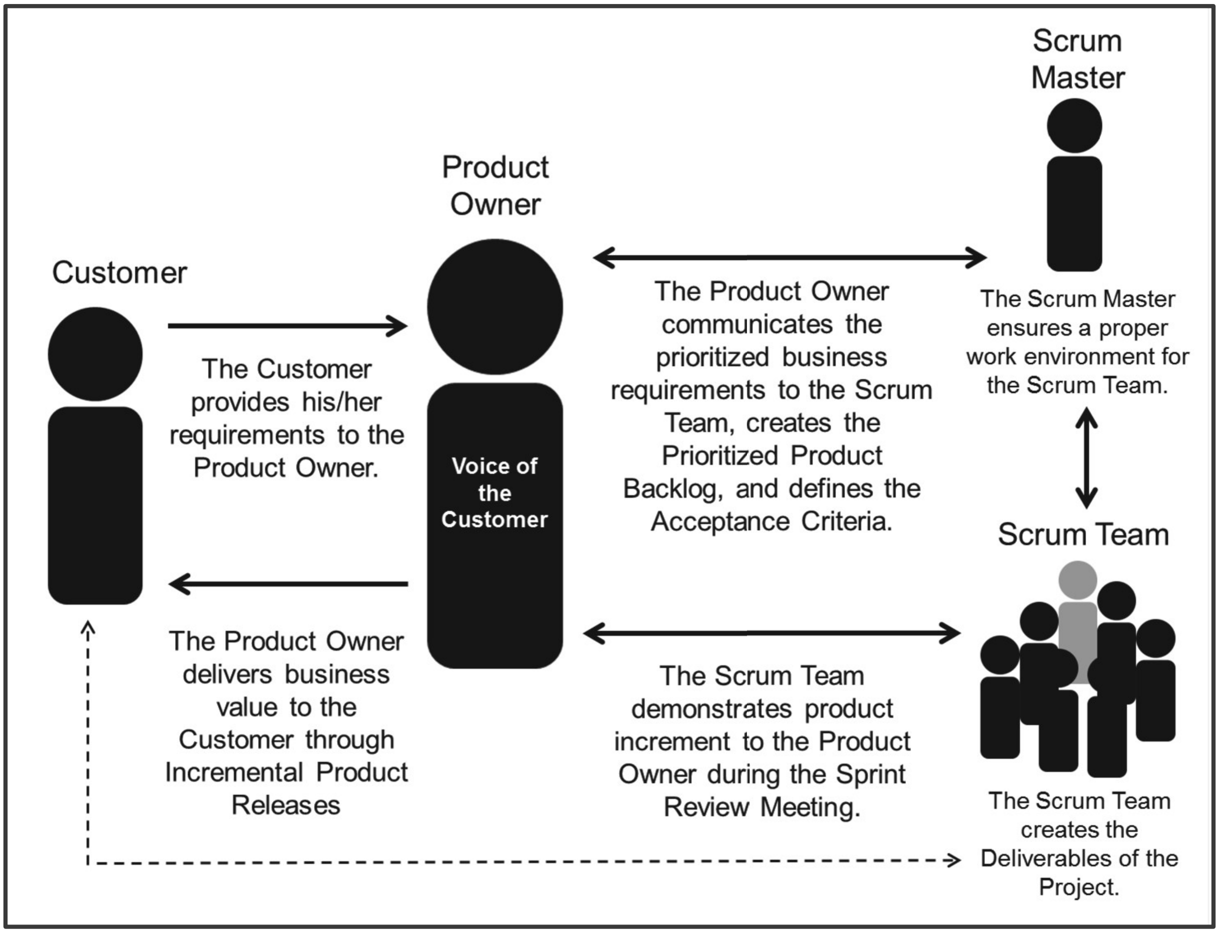
Scrum Master
Product Owner
Customer
The Customer provides his/her requirements to the Product Owner.
Voice of the Customer
The Product Owner communicates the prioritized business requirements to the Scrum Team, creates the Prioritized Product Backlog, and defines the Acceptance Criteria.
The Scrum Master ensures a proper work environment for the Scrum Team.
Scrum Team
The Product Owner delivers business value to the Customer through Incremental Product Releases
The Scrum Team demonstrates product increment to the Product Owner during the Sprint Review Meeting.
The Scrum Team creates the Deliverables of the Project.

Figure 1-4: Organization in Scrum

The Organization aspect of Scrum also addresses the team structure requirements to implement Scrum in programs and portfolios.

1.4.3.2 Business Justification

It is important for an organization to perform a proper business assessment prior to starting any project. This helps key decision makers understand the business need for a change or for a new product or service, the justification for moving forward with a project, and its viability.

Business justification in Scrum is based on the concept of Value-driven Delivery. One of the key characteristics of any project is the uncertainty of results or outcomes. It is impossible to guarantee project success at completion, irrespective of the size or complexity of a project. Considering this uncertainty of achieving success, Scrum attempts to start delivering results as early in the project as possible. This early delivery of results, and thereby value, provides an opportunity for reinvestment and proves the worth of the project to interested stakeholders.

Scrum's adaptability allows the project's objectives and processes to change if its business justification changes. It is important to note that although the Product Owner is primarily responsible for business justification, other team members contribute significantly.

1.4.3.3 Quality

In Scrum, quality is defined as the ability of the completed product or deliverables to meet the Acceptance Criteria and achieve the business value expected by the customer.

To ensure a project meets quality requirements, Scrum adopts an approach of continuous improvement whereby the team learns from experience and stakeholder engagement to constantly keep the Prioritized Product Backlog updated with any changes in requirements. The Prioritized Product Backlog is simply never complete until the closure or termination of the project. Any changes to the requirements reflect changes in the internal and external business environment and allow the team to continually work and adapt to achieve those requirements.

Since Scrum requires work to be completed in increments during Sprints, this means that errors or defects get noticed earlier through repetitive quality testing, rather than when the final product or service is near completion. Moreover, important quality-related tasks (e.g., development, testing, and documentation) are completed as part of the same Sprint by the same team—this ensures that quality is inherent in any deliverable created as part of a Sprint. Such deliverables from Scrum projects, which are potentially shippable, are referred to as 'Done.'

Thus, continuous improvement with repetitive testing optimizes the probability of achieving the expected quality levels in a Scrum project. Constant discussions between the Scrum Core Team and stakeholders (including customers and users) with actual increments of the product being delivered at the end of every Sprint, ensures that the gap between customer expectations from the project and actual deliverables produced is constantly reduced.

The Scrum Guidance Body may also provide guidelines about quality which may be relevant to all Scrum projects in the organization.

1.4.3.4 Change

Every project, regardless of its method or framework used, is exposed to change. It is imperative that project team members understand that the Scrum development processes are designed to embrace change. Organizations should try to maximize the benefits that arise from change and minimize any negative impacts through diligent change management processes in accordance with the principles of Scrum.

A primary principle of Scrum is its acknowledgement that a) stakeholders (e.g., customers, users, and sponsors) change their mind about what they want and need throughout a project (sometimes referred to as "requirements churn") and b) it is very difficult, if not impossible, for stakeholders to define all requirements during project initiation.

Scrum projects welcome change by using short, iterative Sprints that incorporate customer feedback on each Sprint's deliverables. This enables the customer to regularly interact with the Scrum Team members, view deliverables as they are ready, and change requirements if needed earlier in the Sprint.

Also, the portfolio or program management teams can respond to Change Requests pertaining to Scrum projects applicable at their level.

1.4.3.5 Risk

Risk is defined as an uncertain event or set of events that can affect the objectives of a project and may contribute to its success or failure. Risks that are likely to have a positive impact on the project are referred to as opportunities, whereas threats are risks that could affect the project in a negative manner. Managing risk must be done proactively, and it is an iterative process that should begin at project initiation and continue throughout the project's lifecycle. The process of managing risks should follow some standardized steps to ensure that risks are identified, evaluated, and a proper course of action is determined and acted upon accordingly.

Risks should be identified, assessed, and responded to based on two factors: the probability of each risk's occurrence and the possible impact in the event of such occurrence. Risks with a high probability and impact value (determined by multiplying both factors), should be addressed before those with a relatively lower value. In general, once a risk is identified, it is important to understand the risk with regard to the probable causes and the potential effects if the risk occurs.

1.4.4 Scrum Processes

Scrum processes address the specific activities and flow of a Scrum project. In total there are nineteen processes which are grouped into five phases. These phases are presented in chapters 8 through 12 of the *SBOK™ Guide*, as shown in Table 1-1.

Chapter	Phase	Processes
8	Initiate	1. Create Project Vision 2. Identify Scrum Master and Stakeholder(s) 3. Form Scrum Team 4. Develop Epic(s) 5. Create Prioritized Product Backlog 6. Conduct Release Planning
9	Plan and Estimate	7. Create User Stories 8. Approve, Estimate, and Commit User Stories 9. Create Tasks 10. Estimate Tasks 11. Create Sprint Backlog
10	Implement	12. Create Deliverables 13. Conduct Daily Standup 14. Groom Prioritized Product Backlog
11	Review and Retrospect	15. Convene Scrum of Scrums 16. Demonstrate and Validate Sprint 17. Retrospect Sprint
12	Release	18. Ship Deliverables 19. Retrospect Project

Table 1-1: Summary of Scrum Processes

These phases describe each process in detail including their associated inputs, tools, and outputs. In each process, some inputs, tools, and outputs are mandatory (those with an asterisk [*] after their names), while others are optional. Whether to include the optional inputs, tools, and/or outputs depend on the particular project, organization, or industry. Inputs, tools, and outputs denoted as mandatory are important for the successful implementation of Scrum in any organization.

1.4.4.1 Initiate

1. *Create Project Vision*—In this process, the Project Business Case is reviewed to create a Project Vision Statement that will serve as the inspiration and provide focus for the entire project. The Product Owner is identified in this process.

2. *Identify Scrum Master and Stakeholder(s)*—In this process, the Scrum Master and Stakeholders are identified using specific Selection Criteria.

3. *Form Scrum Team*—In this process, Scrum Team members are identified. Normally the Product Owner has the primary responsibility of selecting team members, but often does so in collaboration with the Scrum Master.

4. *Develop Epic(s)*—In this process, the Project Vision Statement serves as the basis for developing Epics. User Group Meetings may be held to discuss appropriate Epics.

5. *Create Prioritized Product Backlog*—In this process, Epic(s) are refined, elaborated, and then prioritized to create a Prioritized Product Backlog for the project. The Done Criteria is also established at this point.

6. *Conduct Release Planning*—In this process, the Scrum Core Team reviews the User Stories in the Prioritized Product Backlog to develop a Release Planning Schedule, which is essentially a phased deployment schedule that can be shared with the project stakeholders. Length of Sprint is also determined in this process.

1.4.4.2 Plan and Estimate

7. *Create User Stories*—In this process, User Stories and their related User Story Acceptance Criteria are created. User Stories are usually written by the Product Owner and are designed to ensure that the customer's requirements are clearly depicted and can be fully understood by all stakeholders. User Story Writing Exercises may be held which involves Scrum Team members creating the User Stories. User Stories are incorporated into the Prioritized Product Backlog.

8. *Approve, Estimate, and Commit User Stories*—In this process, the Product Owner approves User Stories for a Sprint. Then, the Scrum Master and Scrum Team estimate the effort required to develop the functionality described in each User Story, and the Scrum Team commits to deliver the customer requirements in the form of Approved, Estimated, and Committed User Stories.

9. *Create Tasks*—In this process, the Approved, Estimated, and Committed User Stories are broken down into specific tasks and compiled into a Task List. Often a Task Planning Meeting is held for this purpose.

10. *Estimate Tasks*—In this process, the Scrum Core Team, in Task Estimation Meetings, estimate the effort required to accomplish each task in the Task List. The result of this process is an Effort Estimated Task List.

11. *Create Sprint Backlog*—In this process, the Scrum Core Team holds Sprint Planning Meetings where the group creates a Sprint Backlog containing all tasks to be completed in the Sprint.

1.4.4.3 Implement

12. *Create Deliverables*—In this process, the Scrum Team works on the tasks in the Sprint Backlog to create Sprint Deliverables. A Scrumboard is often used to track the work and activities being carried out. Issues or problems being faced by the Scrum Team could be updated in an Impediment Log.

13. *Conduct Daily Standup*—In this process, everyday a highly focused, Time-boxed meeting is conducted referred to as the Daily Standup Meeting. This is the forum for the Scrum Team to update each other on their progress and any impediments they may be facing.

14. *Groom Prioritized Product Backlog*—In this process, the Prioritized Product Backlog is continuously updated and maintained. A Prioritized Product Backlog Review Meeting may be held, in which any changes or updates to the backlog are discussed and incorporated into the Prioritized Product Backlog as appropriate.

1.4.4.4 Review and Retrospect

15. *Convene Scrum of Scrums*—In this process, Scrum Team representatives convene for Scrum of Scrums (SoS) Meetings in predetermined intervals or whenever required to collaborate and track their respective progress, impediments, and dependencies across teams. This is relevant only for large projects where multiple Scrum Teams are involved.

16. *Demonstrate and Validate Sprint*—In this process, the Scrum Team demonstrates the Sprint Deliverables to the Product Owner and relevant stakeholders in a Sprint Review Meeting. The purpose of this meeting is to secure approval and acceptance from the Product Owner for the Deliverables created in the Sprint.

17. *Retrospect Sprint*—In this process, the Scrum Master and Scrum Team meet to discuss the lessons learned throughout the Sprint. This information is documented as lessons learned which can be applied to future Sprints. Often, as a result of this discussion, there may be Agreed Actionable Improvements or Updated Scrum Guidance Body Recommendations.

1.4.4.5 Release

18. *Ship Deliverables*—In this process, Accepted Deliverables are delivered or transitioned to the relevant stakeholders. A formal Working Deliverables Agreement documents the successful completion of the Sprint.

19. *Retrospect Project*—In this process, which completes the project, organizational stakeholders and Scrum Core Team members assemble to retrospect the project and identify, document, and internalize the lessons learned. Often, these lessons lead to the documentation of Agreed Actionable Improvements, to be implemented in future projects.

1.5 Scrum vs. Traditional Project Management

Table 1-2 summarizes many of the differences between Scrum and traditional project management models.

	Scrum	Traditional Project Management
Emphasis is on	People	Processes
Documentation	Minimal—only as required	Comprehensive
Process style	Iterative	Linear
Upfront planning	Low	High
Prioritization of Requirements	Based on business value and regularly updated	Fixed in the Project Plan
Quality assurance	Customer centric	Process centric
Organization	Self-organized	Managed
Management style	Decentralized	Centralized
Change	Updates to Productized Product Backlog	Formal Change Management System
Leadership	Collaborative, Servant Leadership	Command and control
Performance measurement	Business value	Plan conformity
Return on Investment (ROI)	Early/throughout project life	End of project life
Customer involvement	High throughout the project	Varies depending on the project lifecycle

Table 1-2: Scrum vs. Traditional Project Management