SCRUMstudy
Targeting success

A Guide to the

SCRUM BODY OF KNOWLEDGE (SBOK™ GUIDE)

Third Edition

Includes two chapters about Scaling Scrum for Large Projects and the Enterprise

A Comprehensive Guide to Deliver Projects using Scrum

SCRUMstudy

Targeting success

A Guide to the

SCRUM BODY OF KNOWLEDGE

(SBOK™ Guide)

3rd Edition

A Comprehensive Guide to Deliver Projects using Scrum

Library of Congress Cataloging-in-Publication Data

A Guide to the Scrum Body of Knowledge (SBOK™ Guide) –3rd edition

Includes bibliographical references and index.
ISBN: 978-0-9899252-0-4

1. Scrum Framework. I. SCRUMstudy™. II. *SBOK™ Guide*

2013950625

ISBN: 978-0-9899252-0-4

Published by:
SCRUMstudy™, a brand of VMEdu, Inc.
12725 W. Indian School Road, Suite F-112
Avondale, Arizona 85392 USA
Phone: +1-480-882-0706
Fax: +1-240-238-2987
Email: sbok@scrumstudy.com
Website: www.scrumstudy.com

10 9 8 7 6 5 4 3 2

PREFACE

A Guide to the Scrum Body of Knowledge (SBOK™ Guide) provides guidelines for the successful implementation of Scrum—the most popular Agile product development and project delivery approach. Scrum, as defined in the *SBOK™ Guide,* is a framework which is applicable to portfolios, programs, or projects of any size or complexity; and may be applied effectively in *any* industry to create a product, service, or other result.

The *SBOK™ Guide* is intended for use as a reference and knowledge guide by both experienced Scrum and other product or service development practitioners, as well as by persons with no prior experience or knowledge of Scrum or any other project delivery method. This new edition of the *SBOK™ Guide* provides additional insight into Scrum best practices, particularly in the areas of scaling Scrum. Two chapters have been added to the *SBOK™ Guide* to specifically address scaling Scrum for large projects (Chapter 13), and scaling Scrum for the Enterprise (Chapter 14). As the popularity and application of the Scrum framework grows and evolves globally, our goal is to share the lessons learned and best practices as part of the *SBOK™ Guide*.

The *SBOK™ Guide* draws from the combined knowledge and insight gained from thousands of projects across a variety of organizations and industries. This 3rd Edition adds to the collective contributions of experts in Scrum and project delivery. In particular, feedback from the global Scrum community played a large role in identifying improvements and additions to the *SBOK™ Guide*. Its development has truly been a collaborative effort from a large number of experts and practitioners in a variety of disciplines.

Wide adoption of the *SBOK™ Guide* framework standardizes how Scrum is applied to projects across organizations globally, as well as significantly helps to improve their Return on Investment. Additionally, it promotes greater thought and deliberation regarding the application of Scrum to many types of projects, which will in turn contribute towards expanding and enriching the body of knowledge and consequently future updates to this guide.

Although the *SBOK™ Guide* is a comprehensive guide and framework for delivering projects using Scrum, its contents are organized for easy reference, regardless of the reader's prior knowledge on the subject. I hope each reader will learn from and enjoy it as much as the many authors and reviewers learned from and enjoyed the process of collating the collective knowledge and wisdom contained within it.

Tridibesh Satpathy,

Lead Author, *SBOK™ Guide*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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 approach. 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). Chapters 13 and 14 are new additions to the *SBOK™ Guide,* which provide guidance on Scaling Scrum for Large Projects and Scaling Scrum for the Enterprise.

This chapter is divided into the following sections:

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 approach.

Scrum is one of the most popular Agile methods. It is an adaptive, iterative, fast, flexible, and effective framework 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.

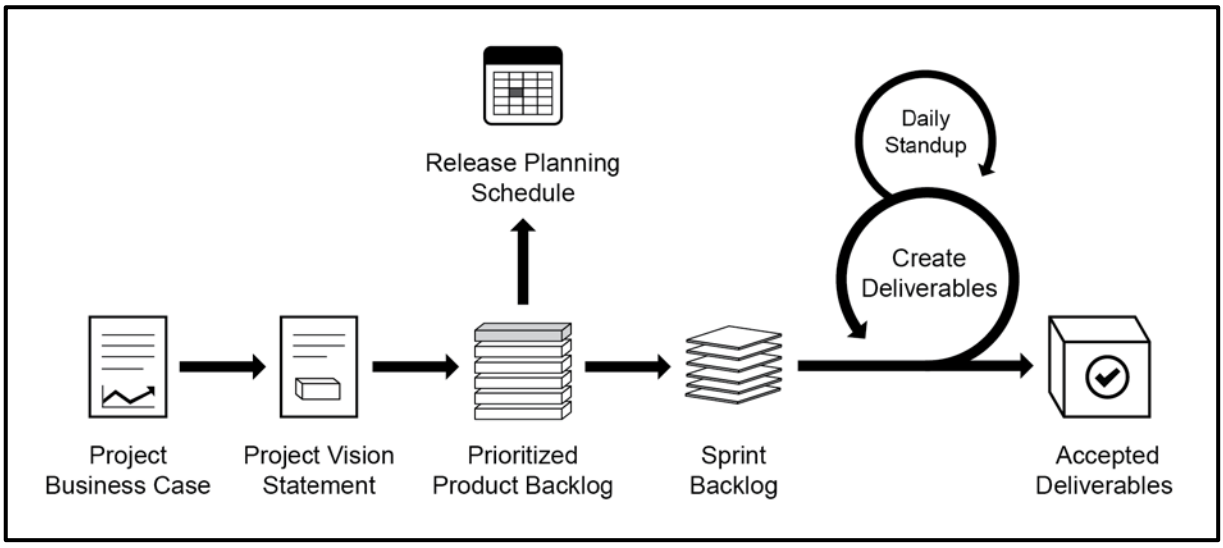


Figure 1-1: Scrum Flow for One Sprint

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 framework. In recent years, Scrum has increased in popularity and is now the preferred project development approach 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 *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, programs, and portfolios. In situations where the Scrum Team size exceeds ten people, multiple Scrum Teams can be formed to work on the project. 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. Large or complex projects are often implemented as part of a program or portfolio.

Details on scaling Scrum for Large Projects is provided in Chapter 13, and scaling Scrum for the Enterprise is covered in Chapter 14.

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™)
- Scaled Scrum Master Certified (SSMC™)
- SCRUMstudy Agile Master Certified (SAMC™)
- Scrum Product Owner Certified (SPOC™)
- Scaled Scrum Product Owner Certified (SSPOC™)
- Expert Scrum Master Certified (ESMC™)

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 fundamental Scrum processes and their associated inputs, tools, and outputs. Chapters 13 and 14 cover the additional processes specific to Scaling Scrum for Large Projects and Scaling Scrum for the Enterprise.

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.

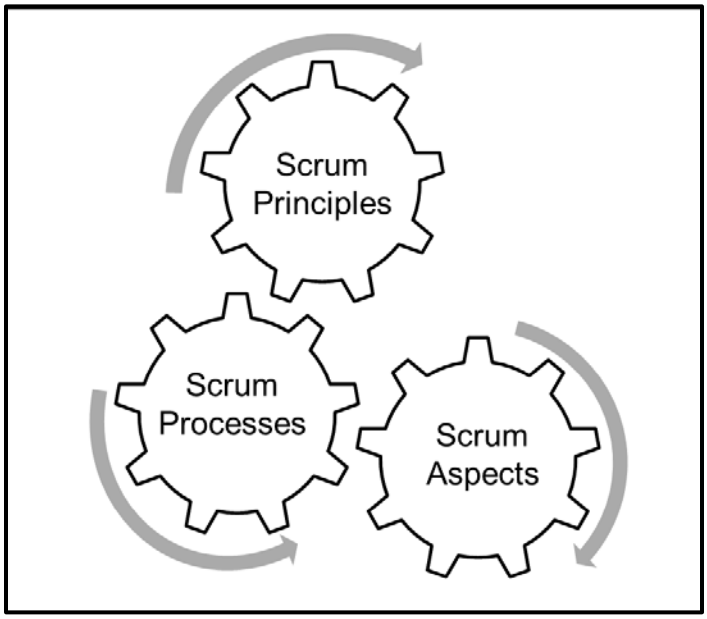


Figure 1-2: SBOK™ Guide Framework

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 approach. 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, or considered critical to success, 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 to benefit from the optional best-practice inputs, tools, and outputs suggested.

Scrum is a framework and is not meant to be prescriptive, which means there is room for flexibility in its application. All the fundamental Scrum processes detailed in the *SBOK™ Guide* (chapters 8 through 12) are required for every Scrum project, but would be applied based on the specific needs of the organization, project, product or team. Additional processes would apply when Scaling Scrum for Large Projects (chapter 13), or Scaling Scrum for the Enterprise (chapter 14).

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.

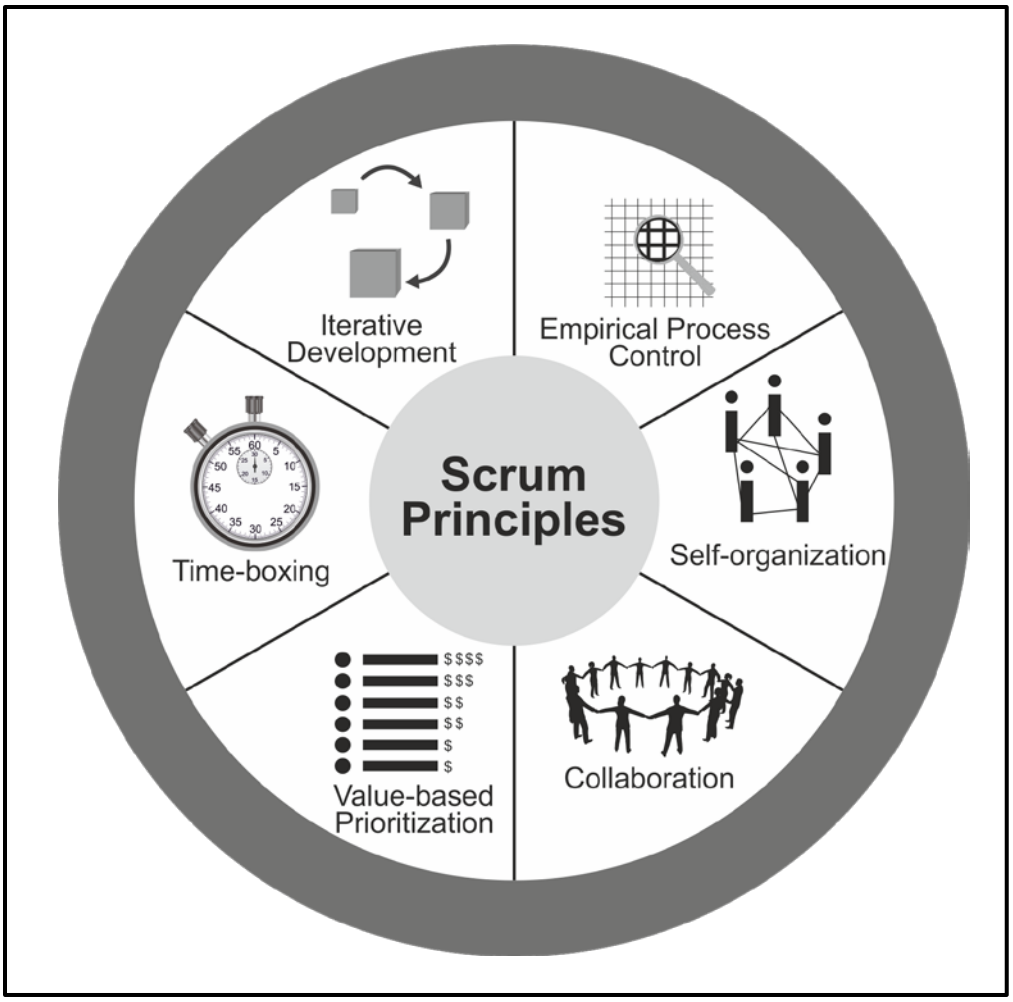


Figure 1-3: Scrum Principles

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.

Figure 1-4 illustrates the Scrum Organization structure.

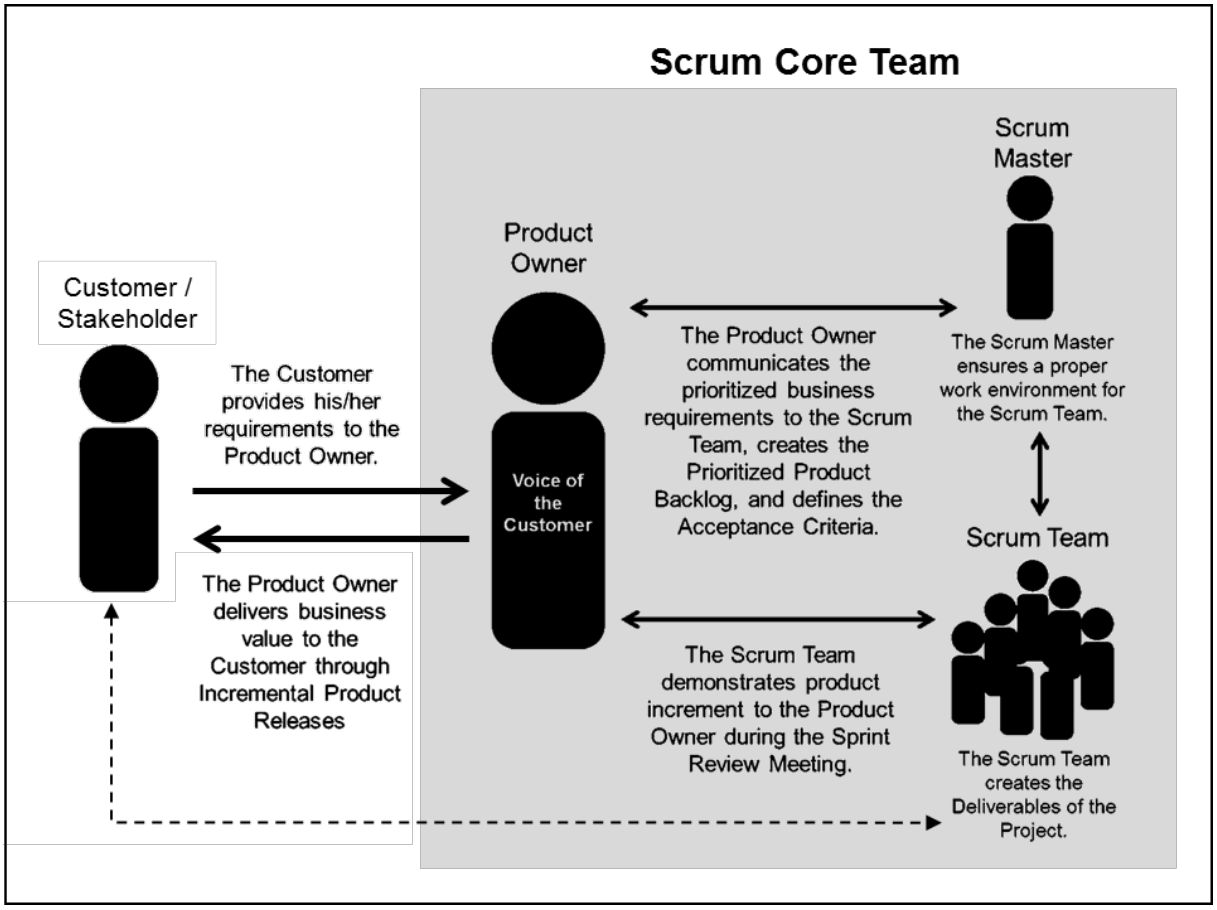


Figure 1-4: Organization in Scrum

The Organization aspect of Scrum also addresses the team structure requirements to implement Scrum in large projects, 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 fundamental Scrum processes that apply to all projects. These processes are grouped into five phases and presented in chapters 8 through 12 of the *SBOK™ Guide*, as shown in Table 1-1.

Chapter	Phase	Fundamental Scrum 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. Estimate User Stories 9. Commit User Stories 10. Identify Tasks 11. Estimate Tasks 12. Create Sprint Backlog
10	Implement	13. Create Deliverables 14. Conduct Daily Standup 15. Groom Prioritized Product Backlog
11	Review and Retrospect	16. Demonstrate and Validate Sprint 17. Retrospect Sprint
12	Release	18. Ship Deliverables 19. Retrospect Project

Table 1-1: Summary of Fundamental 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 depends on the particular project, organization, or industry. Inputs, tools, and outputs denoted with an asterisk are considered mandatory or critical to the successful implementation of Scrum in any organization.

For large scale Scrum projects that require coordination across multiple teams, there are three additional Scrum processes, which are defined in Chapter 13—Scaling Scrum for Large Projects. There are also

specific processes defined when implementing Scrum at the enterprise level, which are covered in Chapter 14—Scaling Scrum for the Enterprise. These additional Scrum processes are summarized in Table 1-2.

Chapter	Applicability	Additional Scrum Processes
13	Scrum for Large Projects	1. Create Large Project Components 2. Conduct and Coordinate Sprints 3. Prepare Large Project Release
14	Scrum for the Enterprise	1. Create Program or Portfolio Components 2. Review and Update Scrum Guidance Body 3. Create and Groom Program or Portfolio Backlog 4. Coordinate Program or Portfolio Components 5. Retrospect Program or Portfolio Releases

Table 1-2: Summary of Additional Scrum Processes

1.4.4.1 Initiate Phase

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 Phase

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. *Estimate User Stories*—In this process, the Product Owner clarifies User Stories in order for the Scrum Master and Scrum Team to estimate the effort required to develop the functionality described in each User Story.

9. *Commit User Stories*—In this process, the Scrum Team commits to deliver Product Owner-approved User Stories for a Sprint. The result of this process would be Committed User Stories.

10. *Identify Tasks*—In this process, the Committed User Stories are broken down into specific tasks and compiled into a Task List.

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

12. *Create Sprint Backlog*—In this process, the Scrum Core Team creates a Sprint Backlog containing all tasks to be completed in a Sprint as part of the Sprint Planning Meeting.

1.4.4.3 Implement Phase

13. *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.

14. *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.

15. *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 Phase

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 Phase

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.

1.4.4.6 Scrum for Large Projects

Create Large Project Components—This process defines how the multiple Product Owners work together and how the multiple Scrum Teams work together. Also, common components and common and specialized resources are identified.

Conduct and Coordinate Sprints— This process addresses specific aspects that should be considered during each Sprint for a large project. If required, Scrum of Scrums Meetings are conducted to coordinate efforts between different Scrum Teams.

Prepare Large Project Release—In some large projects it may make business sense to do a special Sprint prior to a release in order to prepare for releasing the product (to be decided by the project team based on business needs). This process addresses such a preparation Sprint. To have a preparation

Sprint does not mean that any activities that should be completed in the other Sprints can be delayed until this point.

1.4.4.7 Scrum for the Enterprise

Create Program or Portfolio Components—In this process, the Program or Portfolio Product Owner and key stakeholders identify common components and resources required for the program or portfolio. The Minimum Done Criteria are defined and all stakeholders are identified.

Review and Update Scrum Guidance Body—In this process, the Scrum Guidance Body recommendations are regularly reviewed by the Members of the Scrum Guidance Body and are updated when and if necessary. In this process, changes in the membership of the Scrum Guidance Body are also handled.

Create and Groom Program or Portfolio Backlog—In this process, the Program or Portfolio Backlog is created, updated, and maintained. Suggestions for improvements of the Scrum Guidance Body Recommendations may be made and implementation deadlines may be changed based on changed requirements and/or progress of the projects in the program or portfolio.

Coordinate Program or Portfolio Components—In this process, components of the program or portfolio are coordinated. Dependencies between projects are addressed, common impediments are discussed, and best practices are shared. Sometimes, recommendations for improvements of the Scrum Guidance Body are made.

Retrospect Program or Portfolio Releases—In this process, the Program or Portfolio Product Owner and key stakeholders get together to retrospect a program or portfolio release and internalize the lessons learned. Often, these lessons learned lead to Agreed Actionable Improvements to be implemented in future

1.5 Scrum vs. Traditional Project Management

Table 1-3 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 1-3: Scrum vs. Traditional Project Management

2. PRINCIPLES

2.1 Introduction

Scrum principles are the foundation on which the Scrum framework is based. The principles of Scrum can be applied to any type of project or organization, and they must be adhered to in order to ensure appropriate application of Scrum. The aspects and processes of Scrum can be modified to meet the requirements of the project, or the organization using it, but Scrum principles are non-negotiable and must be applied as described in the framework presented in *A Guide to the Scrum Body of Knowledge (SBOK™ Guide).* Keeping the principles intact and using them appropriately instills confidence to the user of the Scrum framework with regard to attaining the objectives of the project. Principles are considered to be the core guidelines for applying the Scrum framework.

Principles, as defined in the *SBOK™ Guide*, are 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 chapter is divided into the following sections:

2.2 Roles Guide—This section outlines which section or subsection is most relevant for each of the core Scrum roles of Product Owner, Scrum Master, and Scrum Team.

2.3 Empirical Process Control—This section describes the first principle of Scrum, and the three main ideas of transparency, inspection, and adaptation.

2.4 Self-organization—This section highlights the second principle of Scrum, which 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.

2.5 Collaboration—This section emphasizes the third principle of Scrum where product development is a shared value-creation process that needs all stakeholders working and interacting together to deliver the greatest value. It also focuses on the core dimensions of collaborative work: awareness, articulation, and appropriation.

2.6 Value-based Prioritization—This section presents the fourth principle of Scrum, which highlights the Scrum framework's drive to deliver maximum business value in a minimum time span.

2.7 Time-boxing—This section explains the fifth principle of Scrum which treats time as a limiting constraint. It also covers the Sprint, Daily Standup Meeting, and the various other Sprint-related meetings such as the Sprint Planning Meeting and Sprint Review Meeting, all of which are Time-boxed.

2.8 Iterative Development—This section addresses the sixth principle of Scrum which emphasizes that iterative development helps to better manage changes and build products that satisfy customer needs.

2.9 Scrum vs. Traditional Project Management—This section highlights the key differences between the Scrum principles and traditional project management (Waterfall model) principles and explains how Scrum works better in today's fast-changing world.

2.2 Roles Guide

All the sections in this chapter are important for all the Scrum Core Team roles—Product Owner, Scrum Master, and Scrum Team. A clear understanding of the Scrum principles by all stakeholders is essential to make Scrum framework a success in any organization.

2.3 Empirical Process Control

In Scrum, decisions are made based on observation and experimentation rather than on detailed upfront planning. Empirical process control relies on the three main ideas of transparency, inspection, and adaptation.

2.3.1 Transparency

Transparency allows all facets of any Scrum process to be observed by anyone. This promotes an easy and transparent flow of information throughout the organization and creates an open work culture. In Scrum, transparency is depicted through the following:

- A Project Vision Statement which can be viewed by all stakeholders and the Scrum Team
- An open Prioritized Product Backlog with prioritized User Stories that can be viewed by everyone, both within and outside the Scrum Team
- A Release Planning Schedule which may be coordinated across multiple Scrum Teams
- Clear visibility into the team's progress through the use of a Scrumboard, Burndown Chart, and other information radiators
- Daily Standup Meetings conducted during the *Conduct Daily Standup* process, in which all team members report what they have done the previous day, what they plan to do today, and any problems preventing them from completing their tasks in the current Sprint

- Sprint Review Meetings conducted during the *Demonstrate and Validate Sprint* process, in which the Scrum Team demonstrates the potentially shippable Sprint Deliverables to the Product Owner and Stakeholders

Figure 2-1 summarizes the concept of transparency in Scrum.

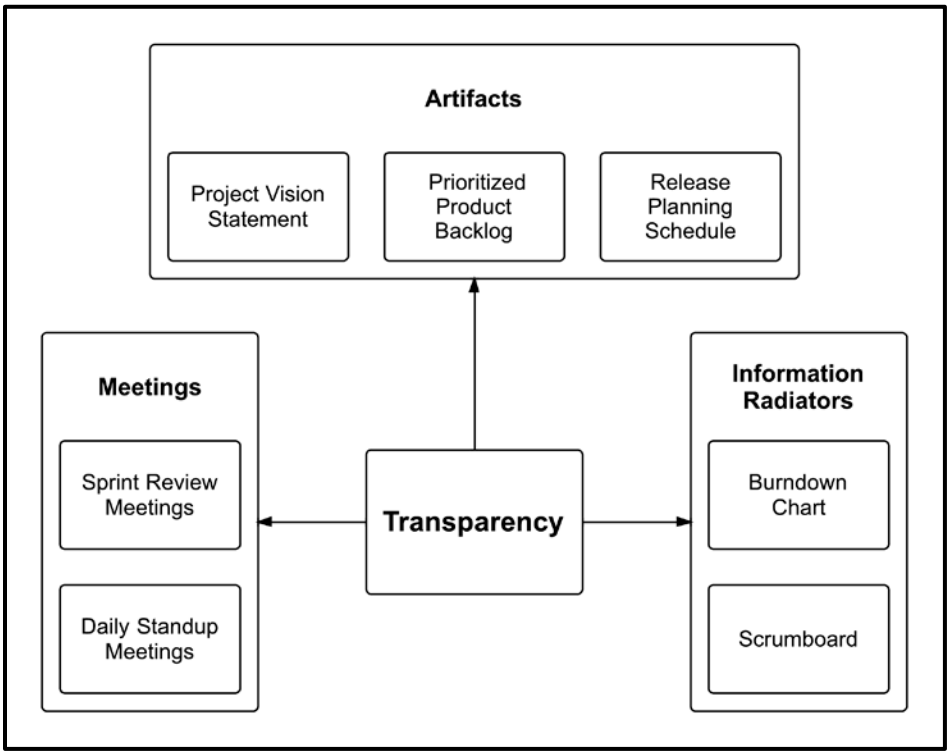


Figure 2-1: Transparency in Scrum

2.3.2 Inspection

Inspection in Scrum is depicted through the following:

- Use of a common Scrumboard and other information radiators which show the progress of the Scrum Team on completing the tasks in the current Sprint.
- Collection of feedback from the customer and other stakeholders during the *Develop Epic(s)*, *Create Prioritized Product Backlog,* and *Conduct Release Planning* processes.
- Inspection and approval of the Deliverables by the Product Owner and the customer in the *Demonstrate and Validate Sprint* process.

Figure 2-2 summarizes the concept of inspection in Scrum.

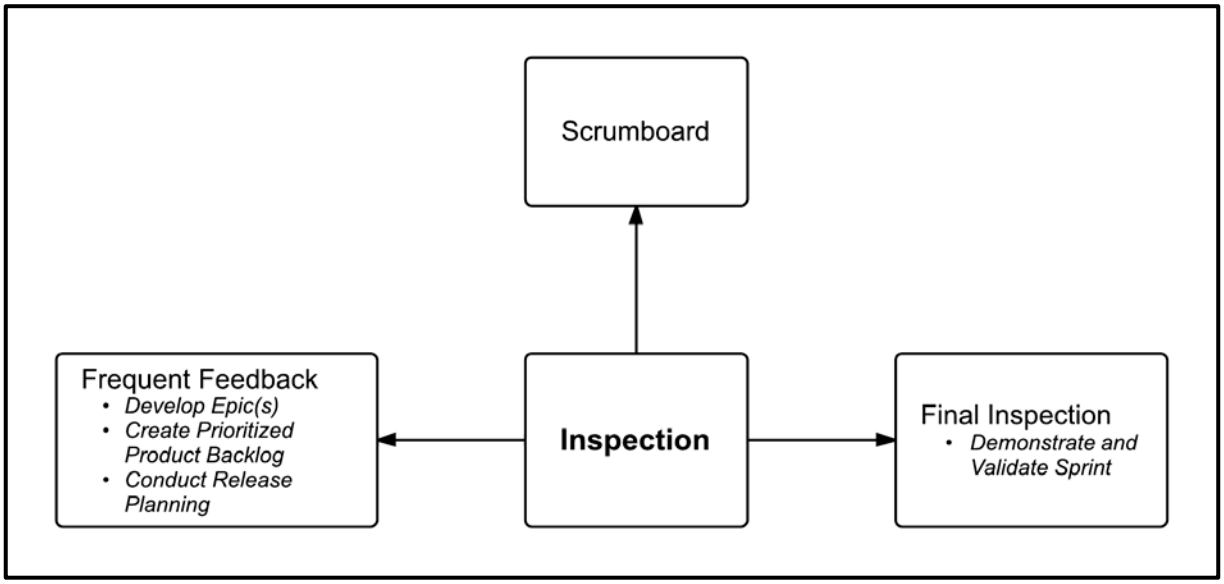


Figure 2-2: Inspection in Scrum

2.3.3 Adaptation

Adaptation happens as the Scrum Core Team and Stakeholders learn through transparency and inspection and then adapt by making improvements in the work they are doing. Some examples of adaptation include:

- In Daily Standup Meetings, Scrum Team members openly discuss impediments to completing their tasks and seek help from other team members. More experienced members in the Scrum Team also mentor those with relatively less experience in knowledge of the project or technology.
- Risk identification is performed and iterated throughout the project. Identified risks become inputs to several Scrum processes including *Create Prioritized Product Backlog, Groom Prioritized Product Backlog,* and *Demonstrate and Validate Sprint*.

- Improvements can also result in Change Requests, which are discussed and approved during the *Develop Epic(s)*, *Create Prioritized Product Backlog,* and *Groom Prioritized Product Backlog* processes.
- The Scrum Guidance Body interacts with Scrum Team members during the *Create User Stories*, *Estimate Tasks*, *Create Deliverables*, and *Groom Prioritized Product Backlog* processes to offer guidance and also provide expertise as required.
- In the *Retrospect Sprint* process, Agreed Actionable Improvements are determined based on the outputs from the *Demonstrate and Validate Sprint* process.
- In Retrospect Project Meeting, participants document lessons learned and perform reviews looking for opportunities to improve processes and address inefficiencies.

Figure 2-3 summarizes the concept of adaptation in Scrum.

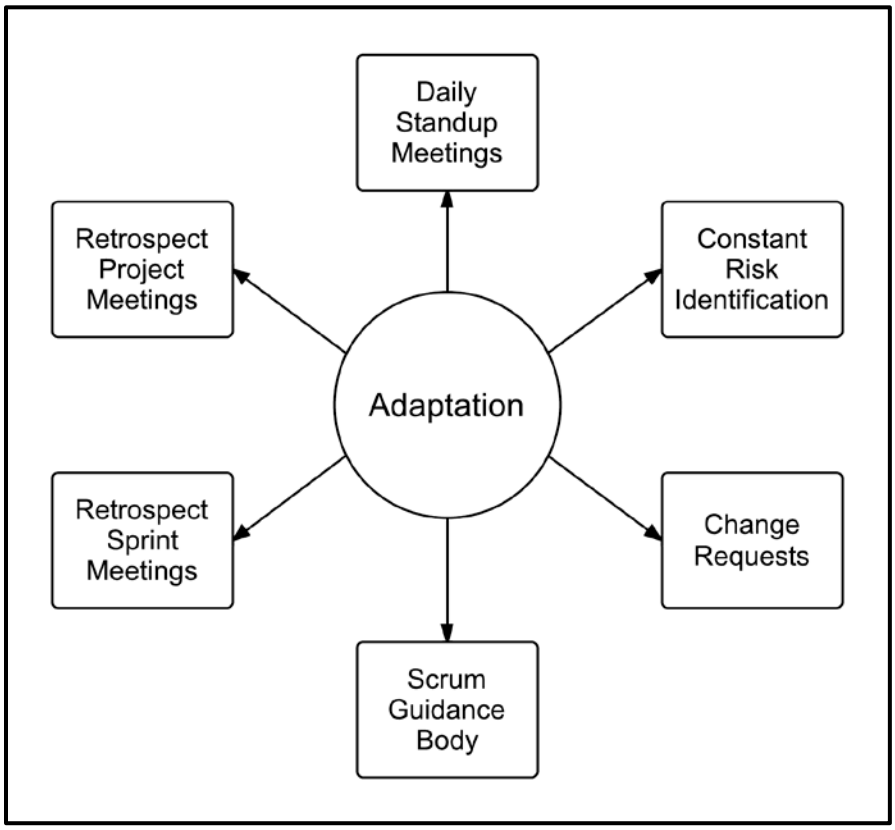


Figure 2-3: Adaptation in Scrum

With other methods, like the traditional Waterfall model, considerable planning needs to be done in advance and the customer generally does not review product components until near the end of a phase, or the end of the entire project. This method often presents huge risks to the project's success because it may have more potential for significantly impacting project delivery and customer acceptance. The customer's interpretation

and understanding of the finished product may be very different from what was actually understood and produced by the team and this may not be known until very late in the project's development.

Figure 2-4 demonstrates an example of these challenges.

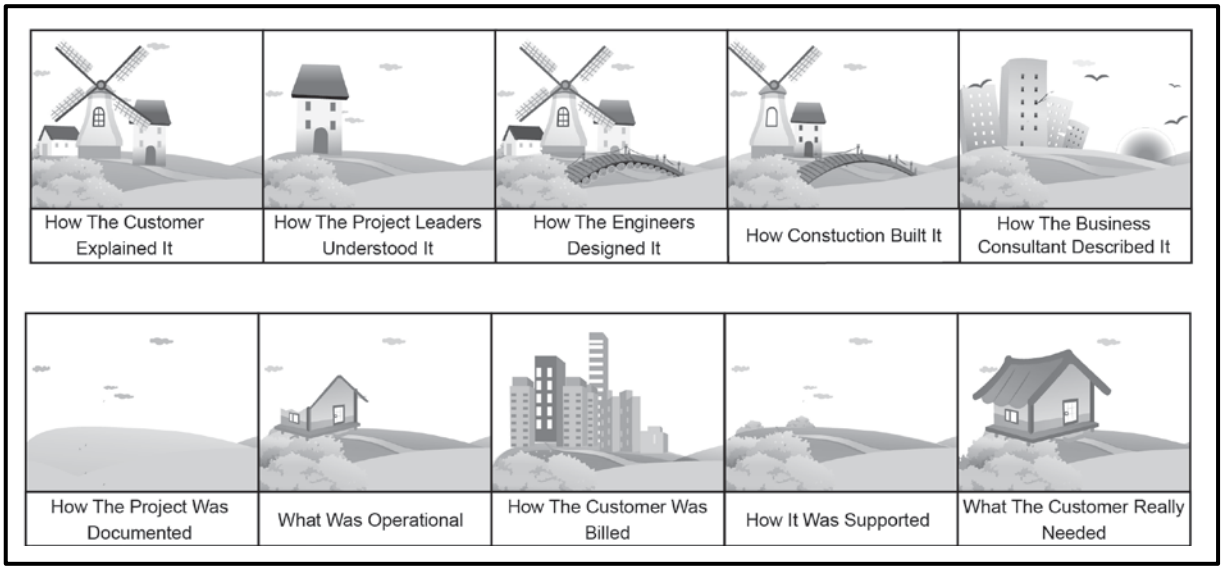


Figure 2-4: Challenges in Traditional Project Management

2.4 Self-organization

Scrum believes that employees are self-motivated and seek to accept greater responsibility. So, they deliver much greater value when self-organized.

The preferred leadership style in Scrum is "servant leadership", which emphasizes achieving results by focusing on the needs of the Scrum Team. See section 3.10.3 for a discussion of various leadership and management styles.

2.4.1 Benefits of Self-organization

Self-organization as an essential principle in Scrum leads to the following:

- Team buy-in and shared ownership
- Motivation, which leads to an enhanced performance level of the team
- Innovative and creative environment conducive to growth

Self-organization does not mean that team members are allowed to act in any manner that they want to. It just means that once the Product Vision is defined in the *Create Project Vision* process, the Product Owner, Scrum Master, and Scrum Team get identified. Also the Scrum Core Team itself works very closely with relevant Stakeholder(s) for refining requirements better as they go through the *Develop Epic(s)* and *Create User Stories* process. Team expertise is used to assess the inputs needed to execute the planned work of the project. This judgment and expertise are applied to all technical and management aspects of the project during the *Create Deliverables* process.

Although prioritization is primarily done by the Product Owner who represents the Voice of Customer, the self-organized Scrum Team is involved in task breakdown and estimation during the *Identify Tasks* and *Estimate Tasks* processes. During these processes, each team member is responsible for determining what work he or she will be doing. During the execution of a Sprint, if team members need any help with completing their tasks, Scrum addresses this through the regular interaction mandatory with the Daily Standup Meetings. The Scrum Team itself interacts with other teams through the Scrum of Scrums (SoS) Meetings and can look for additional guidance as required from the Scrum Guidance Body.

Finally, the Scrum Team and Scrum Master work closely to demonstrate the product increment created during the Sprint in the *Demonstrate and Validate Sprint* process where properly completed deliverables are accepted. Since the Deliverables are potentially shippable, (and the Prioritized Product Backlog is prioritized by User Stories in the order of value created by them), the Product Owner and the customer can clearly visualize and articulate the value being created after every Sprint; and Scrum Teams in turn have the satisfaction of seeing their hard work being accepted by the customer and other stakeholders.

The chief goals of self-organizing teams are as follows:

- Understand the Project Vision and why the project delivers value to the organization
- Estimate User Stories during the *Estimate User Stories* process and assign tasks to themselves during the *Create Sprint Backlog* process
- Identify tasks independently during the *Identify Tasks* process
- Apply and leverage their expertise from being a cross-functional team to work on the tasks during the *Create Deliverables* process
- Deliver tangible results which are accepted by the customer and other stakeholders during the *Demonstrate and Validate Sprint* process
- Resolve individual problems together by addressing them during Daily Standup Meetings
- Clarify any discrepancies or doubts and be open to learning new things
- Upgrade knowledge and skill on a continuous basis through regular interactions within the team
- Maintain stability of team members throughout the duration of the project by not changing members, unless unavoidable

Figure 2-5 illustrates the goals of a self-organizing team.

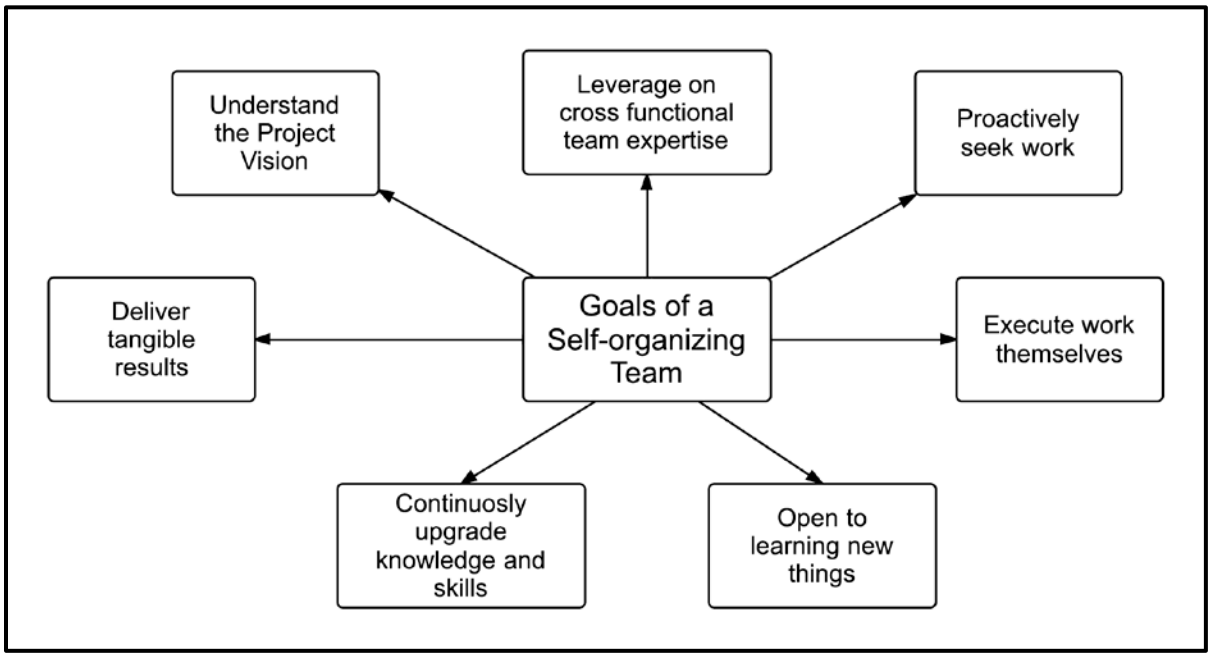


Figure 2-5: Goals of a Self-Organizing Team

2.5 Collaboration

Collaboration in Scrum refers to the Scrum Core Team working together and interfacing with the stakeholders to create and validate the deliverables of the project to meet the goals outlined in the Project Vision. It is important to note the difference between cooperation and collaboration here. Cooperation occurs when the work product consists of the sum of the work efforts of various people on a team. Collaboration occurs when a team works together to play off each other's inputs to produce something greater.

The core dimensions of collaborative work are as follows:

- *Awareness*—Individuals working together need to be aware of each other's work.
- *Articulation*—Collaborating individuals must partition work into units, divide the units among team members, and then after the work is done, reintegrate it.
- *Appropriation*—Adapting technology to one's own situation; the technology may be used in a manner completely different than expected by the designers.

2.5.1 Benefits of Collaboration in Scrum Projects

The Agile Manifesto (Fowler & Highsmith, 2001) stresses "customer collaboration over contract negotiation." Thus, the Scrum framework adopts an approach in which the Scrum Core Team members (Product Owner, Scrum Master, and Scrum Team), collaborate with each other and the stakeholders to create the deliverables that provide greatest possible value to the customer. This collaboration occurs throughout the project.

Collaboration ensures that the following project benefits are realized:

1. The need for changes due to poorly clarified requirements is minimized. For example, during the *Create Project Vision, Develop Epic(s)*, and *Create Prioritized Product Backlog* processes, the Product Owner collaborates with stakeholders to create the Project Vision, Epic(s), and Prioritized Product Backlog, respectively. This will ensure that there is clarity among Scrum Core Team members on the work that is required to complete the project. The Scrum Team collaborates continuously with the Product Owner and stakeholders through a transparent Prioritized Product Backlog to create the project deliverables. The processes *Conduct Daily Standup*, *Groom Prioritized Product Backlog,* and *Retrospect Sprint* provide scope to the Scrum Core Team members to discuss what has been done and collaborate on what needs to be done. Thus the number of Change Requests from the customer and rework is minimized.

2. Risks are identified and dealt with efficiently. For example, risks to the project are identified and assessed in the *Develop Epic(s), Create Deliverables,* and *Conduct Daily Standup* processes by the Scrum Core Team members. The Scrum meeting tools such as the Daily Standup Meeting, Sprint Planning Meeting, Prioritized Product Backlog Review Meeting, and so on provide opportunities to

the team to not only identify and assess risks, but also to implement risk responses to high-priority risks.

3. True potential of the team is realized. For example, the *Conduct Daily Standup* process provides scope for the Scrum Team to collaborate and understand the strengths and weaknesses of its members. If a team member has missed a task deadline, the Scrum Team members align themselves collaboratively to complete the task and meet the targets agreed to for completing the Sprint.

4. Continuous improvement is ensured through lessons learned. For example, the Scrum Team uses the *Retrospect Sprint* process to identify what went well and what did not go well in the previous Sprint. This provides an opportunity to the Scrum Master to work with the team to rework and improve the team for the next scheduled Sprint. This will also ensure that collaboration is even more effective in the next Sprint.

Figure 2-6 illustrates the benefits of collaboration in Scrum projects.

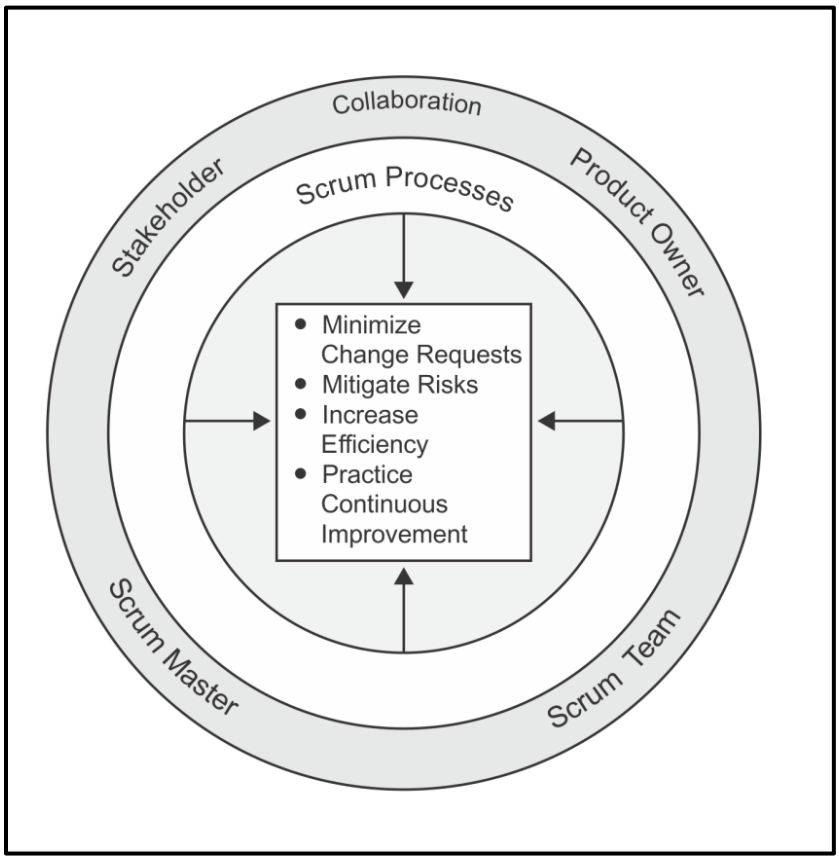


Figure 2-6: Benefits of Collaboration in Scrum Projects

2.5.2 Importance of Colocation in Collaboration

For many of the Scrum practices, high-bandwidth communication is required. To enable this, it is preferred that team members are colocated. Colocation allows both formal and informal interaction between team members. This provides the advantage of having team members always at hand for coordination, problem-solving, and learning. Some of the benefits of colocation are the following:

- Questions get answered quickly.
- Problems are fixed on the spot.
- Less friction occurs between interactions.
- Trust is gained and awarded much more quickly.

Collaboration tools that can be used for colocated or distributed teams are as follows:

1. Colocated Teams (i.e., teams working in the same office)—In Scrum, it is preferable to have colocated teams. If colocated, preferred modes of communication include face-to-face interactions, Decision Rooms or War Rooms, Scrumboards, wall displays, shared tables, and so on.

2. Distributed Teams (i.e., teams working in different physical locations)—Although colocated teams are preferred, at times the Scrum Team may be distributed due to outsourcing, offshoring, different physical locations, work-from-home options, etc. Some tools that could be used for effective collaboration with distributed teams include video conferencing, instant messaging, chats, social media, shared screens, and software tools which simulate the functionality of Scrumboards, wall displays, and so on.

2.6 Value-based Prioritization

The Scrum framework is driven by the goal of delivering maximum business value in a minimum time span. One of the most effective tools for delivering the greatest value in the shortest amount of time is prioritization.

Prioritization can be defined as determination of the order and separation of what must be done now, from what needs to be done later. The concept of prioritization is not new to project management. The traditional Waterfall model of project management proposes using multiple task prioritization tools. From the Project Manager's point of view, prioritization is integral because certain tasks must be accomplished first to expedite the development process and achieve the project goals. Some of the traditional techniques of task prioritization include setting deadlines for delegated tasks and using prioritization matrices.

Scrum, however, uses Value-based Prioritization as one of the core principles that drives the structure and functionality of the entire Scrum framework—it helps projects benefit through adaptability and iterative development of the product or service. More significantly, Scrum aims at delivering a valuable product or service to the customer on an early and continuous basis.

Prioritization is done by the Product Owner when he or she prioritizes User Stories in the Prioritized Product Backlog. The Prioritized Product Backlog contains a list of all the requirements needed to bring the project to fruition.

Once the Product Owner has received the business requirements from the customer and written these down in the form of workable User Stories, he or she works with the customer and sponsor to understand which business requirements provide maximum business value. The Product Owner must understand what the customer wants and values in order to arrange the Prioritized Product Backlog Items (User Stories) by relative importance. Sometimes, a customer may mandate all User Stories to be of high priority. While this might be true, even a list of high-priority User Stories needs to be prioritized within the list itself. Prioritizing a backlog involves determining the criticality of each User Story. High-value requirements are identified and moved to the top of the Prioritized Product Backlog. The processes in which the principle of Value-based Prioritization is put into practice are *Create Prioritized Product Backlog* and *Groom Prioritized Product Backlog.*

Simultaneously, the Product Owner must work with the Scrum Team to understand the project risks and uncertainty as they may have negative consequences associated with them. This should be taken into account while prioritizing User Stories on a value-based approach (refer to the Risk chapter for more information). The Scrum Team also alerts the Product Owner of any dependencies that arise out of implementation. These dependencies must be taken into account during prioritization. Prioritization may be based on a subjective estimate of the projected business value or profitability, or it can be based on results and analysis of the market using tools including, but not limited to, customer interviews, surveys, and financial models and analytical techniques.

The Product Owner has to translate the inputs and needs of the project stakeholders to create the Prioritized Product Backlog. Hence, while prioritizing the User Stories in the Prioritized Product Backlog, the following three factors are considered (see Figure 2-7):

1. Value
2. Risk or uncertainty
3. Dependencies

Thus prioritization results in deliverables that satisfies the requirements of the customer with the objective of delivering the maximum business value in the least amount of time.

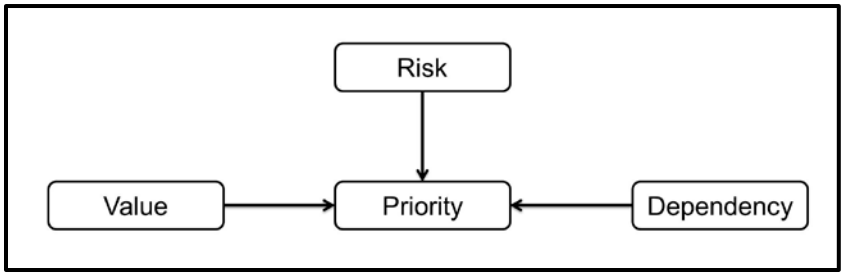


Figure 2-7: Value-based Prioritization

2.7 Time-boxing

Scrum treats time as one of the most important constraints in managing a project. To address the constraint of time, Scrum introduces a concept called 'Time-boxing' which proposes fixing a certain amount of time for each process and activity in a Scrum project. This ensures that Scrum Team members do not take up too much or too little work for a particular period of time and do not expend their time and energy on work for which they have little clarity.

Some of the advantages of Time-boxing are as follows:

- Efficient development process
- Less overheads
- High velocity for teams

Time-boxing can be utilized in many Scrum processes, for example, in the *Conduct Daily Standup* process, the duration of the Daily Standup Meeting is Time-boxed. At times, Time-boxing may be used to avoid excessive improvement of an item (i.e., gold-plating).

Time-boxing is a critical practice in Scrum and should be applied with care. Arbitrary Time-boxing can lead to de-motivation of the team and may have the consequence of creating an apprehensive environment, so it should be used appropriately.

2.7.1 Scrum Time-boxes

- Sprint—A Sprint is a Time-boxed iteration of one to six weeks in duration during which the Scrum Master guides, facilitates, and shields the Scrum Team from both internal and external impediments during the *Create Deliverables* process. This aids in avoiding vision creep that could affect the Sprint goal. During this time, the team works to convert the requirements in the Prioritized Product Backlog into shippable product functionalities. To get maximum benefits from a Scrum project, it is always recommended to keep the Sprint Time-boxed to 4 weeks, unless there are projects with very stable requirements, where Sprints can extend up to 6 weeks.

- Daily Standup Meeting—The Daily Standup Meeting is a short daily meeting, Time-boxed to 15 minutes. The team members get together to report the progress of the project by answering the following three questions:

    1. What have I done since the last meeting?
    2. What do I plan to do before the next meeting?
    3. What impediments or obstacles (if any) am I currently facing?

    This meeting is carried out by the team as part of the *Conduct Daily Standup* process.

- Sprint Planning Meeting—This meeting is conducted prior to the Sprint as part of the *Commit User Stories, Identify Tasks, Estimate Tasks, and Create Sprint Backlog* processes. It is Time-boxed to eight hours for a one-month Sprint. The Sprint Planning Meeting is divided into two parts:

    1. Objective Definition—During the first part of the meeting, the Product Owner explains the highest priority User Stories or requirements in the Prioritized Product Backlog to the Scrum Team. The Scrum Team in collaboration with the Product Owner then commits to the User Stories, which define the Sprint goal.

    2. Task Identification and Estimation—The Scrum Team then decides "how" to complete the selected Prioritized Product Backlog Items to fulfill the Sprint goal. The Committed User Stories and related Effort Estimated Tasks are included in the Sprint Backlog to be tracked.

- Sprint Review Meeting—The Sprint Review Meeting is Time-boxed to four hours for a one-month Sprint. During the Sprint Review Meeting that is conducted in the *Demonstrate and Validate Sprint* process, the Scrum Team presents the deliverables of the current Sprint to the Product Owner. The Product Owner reviews the product (or product increment) against the agreed Acceptance Criteria and either accepts or rejects the completed User Stories.

- Retrospect Sprint Meeting—The Retrospect Sprint Meeting is Time-boxed to 4 hours for a one-month Sprint and conducted as part of the *Retrospect Sprint* process. During this meeting, the Scrum Team gets together to review and reflect on the previous Sprint in terms of the processes followed, tools employed, collaboration and communication mechanisms, and other aspects relevant to the project. The team discusses what went well during the previous Sprint and what did not go well, the goal being to learn and make improvements in the Sprints to follow. Some improvement opportunities or best practices from this meeting could also be updated as part of the Scrum Guidance Body documents.

  Figure 2-8 illustrates the Time-boxed durations for Scrum-related meetings.

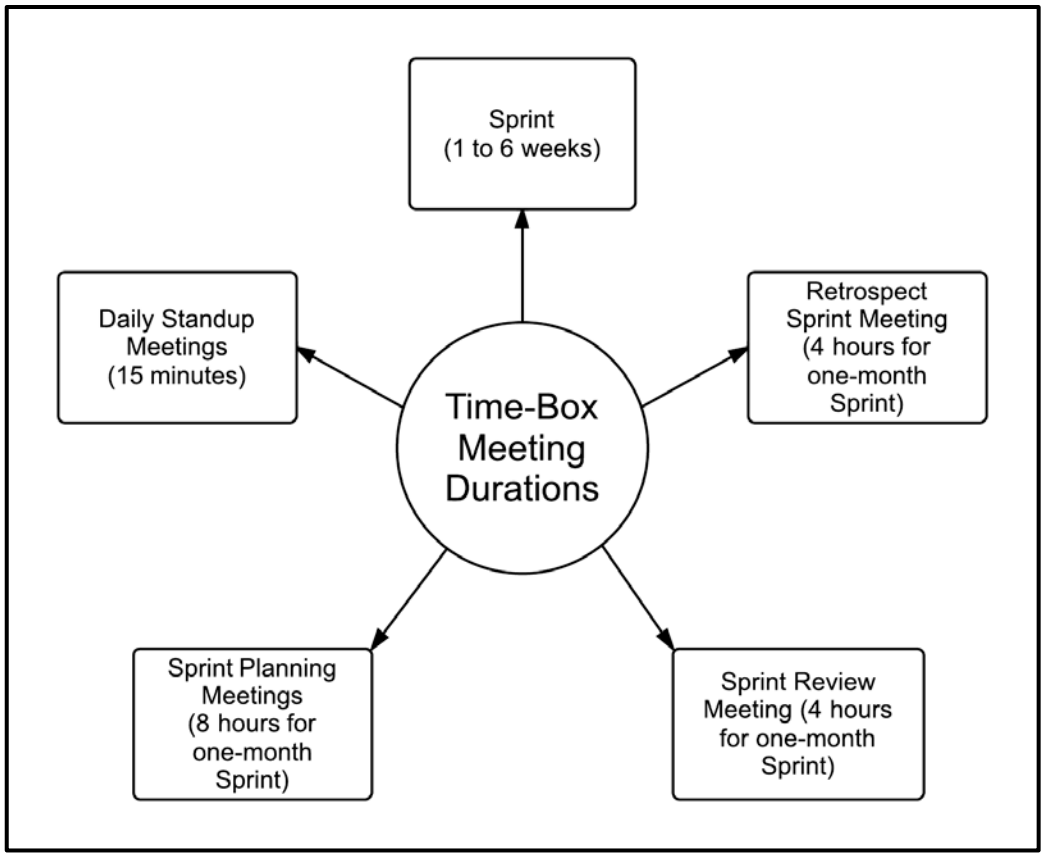


Figure 2-8: Time-Box Durations for Scrum Meetings

2.8 Iterative Development

The Scrum framework is driven by the goal of delivering maximum business value in a minimum time span. To achieve this practically, Scrum believes in Iterative Development of Deliverables.

In most complex projects, the customer may not be able to define very concrete requirements or is not confident of what the end product may look like. The iterative model is more flexible in ensuring that any change requested by the customer can be included as part of the project. User Stories may have to be written constantly throughout the duration of the project. In the initial stages of writing, most User Stories are high-level functionalities. These User Stories are known as Epic(s). Epic(s) are usually too large for teams to complete in a single Sprint. Therefore, they are broken down into smaller User Stories.

Each complex aspect of the project is broken down through progressive elaboration during the *Groom Prioritized Product Backlog* process. The *Create User Stories* and the *Estimate, Approve, and Commit User Stories* processes are used to add new requirements to the Prioritized Product Backlog. The Product Owner's task is to ensure increased ROI by focusing on value and its continuous delivery with each Sprint. The Product Owner should have a very good understanding of the project's business justification and the value the project is supposed to deliver as he drafts the Prioritized Product Backlog and thereby decides what deliverables and hence values are delivered in each Sprint. Then the *Identify Tasks, Estimate Tasks,* and *Create Sprint Backlog* processes produce the Sprint Backlog which the team uses to create the deliverables.

In each Sprint, the *Create Deliverables* process is used to develop the Sprint's outputs. The Scrum Master has to ensure that the Scrum processes are followed and facilitates the team to work in the most productive manner possible. The Scrum Team self-organizes and aims to create the Sprint Deliverables from the User Stories in the Sprint Backlog. In large projects, various cross-functional teams work in parallel across Sprints, delivering potentially shippable solutions at the end of each Sprint. After the Sprint is complete, The Product Owner accepts or rejects the deliverables based on the Acceptance Criteria in the *Demonstrate and Validate Sprint* process.

As illustrated in Figure 2-9, Scrum projects are completed in an iterative manner delivering value throughout the lifecycle of the project.

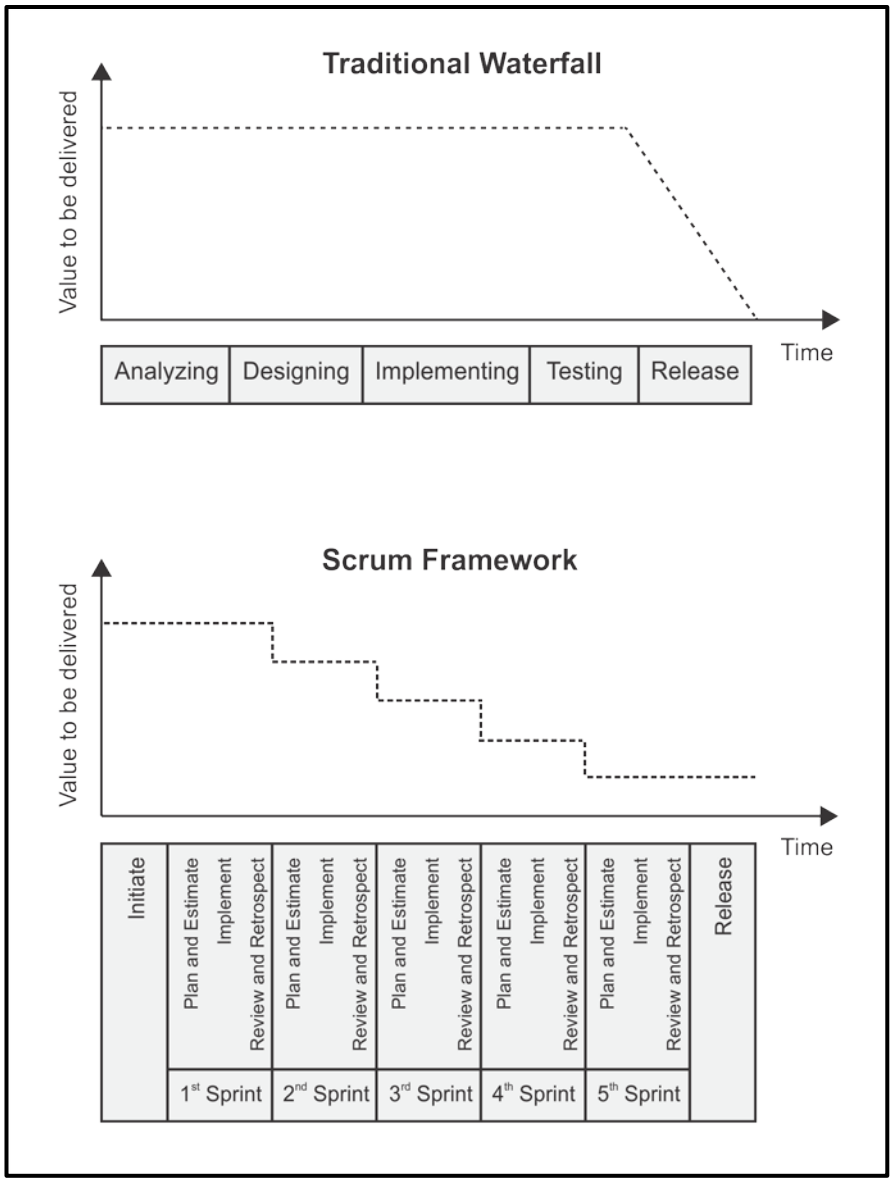


Figure 2-9: Scrum vs. Traditional Waterfall

The benefit of iterative development is that it allows for course correction as all the people involved get better understanding of what needs to be delivered as part of the project and incorporate these learning in an iterative manner. Thus, the time and effort required to reach the final end point is greatly reduced and the team produces deliverables that are better suited to the final business environment.

2.9 Scrum vs. Traditional Project Management

The emphasis in traditional Project Management is to conduct detailed upfront planning for the project with emphasis on fixing the scope, cost, and schedule and managing those parameters. Traditional project management may at times lead to a situation where the plan has succeeded yet the customer is not satisfied.

The Scrum Framework is founded on the belief that the knowledge workers of today can offer much more than just their technical expertise, and that trying to fully map out and plan for an ever-changing environment is not efficient. Therefore, Scrum encourages data-based, iterative decision making. In Scrum, the primary focus is on delivering products that satisfy customer requirements in small iterative shippable increments.

To deliver the greatest amount of value in the shortest amount of time, Scrum promotes prioritization and Time-boxing over fixing the scope, cost and schedule of a project. An important feature of Scrum is self-organization, which allows the individuals who are actually doing the work to estimate and take ownership of tasks.

3. ORGANIZATION

3.1 Introduction

In this section, we will be discussing the various facets of a Scrum project organization as well as core and non-core roles and how to form high performance Scrum Teams.

Organization, as defined in *A Guide to the Scrum Body of Knowledge (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 chapter is divided into the following sections:

3.2 Roles Guide—This section identifies which section or subsection is important for a Product Owner, Scrum Master, and Scrum Team.

3.3 Scrum Project Roles—This section covers all the key core and non-core roles associated with a Scrum project.

3.4 Product Owner—This section highlights the key responsibilities of the Product Owner in relation to a Scrum project, program, and portfolio.

3.5 Scrum Master—This section focuses on the key responsibilities of the Scrum Master in the context of a Scrum project, program, and portfolio.

3.6 Scrum Team—This section emphasizes the key responsibilities of the Scrum Team in the context of a Scrum project.

3.7 Scrum in Projects, Programs, and Portfolios—This section focuses on how the Scrum framework can be tailored and used in the different contexts of programs and portfolios. It also highlights the specific responsibilities of the Scrum Team members in relation to communication, integration, and working with the corporate and program management teams.

3.8 Responsibilities—This section describes the responsibilities relevant to the Organization theme, for everyone working on a project, based on their roles.

3.9 Scrum vs. Traditional Project Management—This section explains the key differences and advantages of the Scrum model in relation to the traditional Waterfall model of project management.

3.10 Popular HR Theories and their Relevance to Scrum—This section contains some of the most popular HR theories useful for all the members in the Scrum Core Team.

3.2 Roles Guide

1. Product Owner—It is imperative for Product Owners to read the entire chapter.
2. Scrum Master—The Scrum Master should also be familiar with this entire chapter with primary focus on sections 3.3, 3.5, 3.6, 3.8 and 3.10.4.
3. Scrum Team— The Scrum Team should mainly focus on sections 3.3, 3.6, and 3.8.

3.3 Scrum Project Roles

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

Scrum roles fall into two broad categories:

1. Core Roles—Core roles are those roles which are mandatorily required for producing the product of the project, are committed to the project, and ultimately are responsible for the success of each Sprint of the project and of the project as a whole.

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, have no formal role on the project team, may interface with the team, but may not be responsible for the success of the project. The non-core roles should also be taken into account in any Scrum project.

3.3.1 Core Roles

There are three core roles in Scrum that are ultimately responsible for meeting the project objectives. The core roles are the Product Owner, Scrum Master, and Scrum Team. Together they are referred to as the Scrum Core Team. It is important to note that, of these three roles, no role has authority over the others.

1. Product Owner

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

Corresponding to a Product Owner role in a project, there could be a Program Product Owner for a program or a Portfolio Product Owner for a portfolio.

2. Scrum Master

The Scrum Master is a facilitator who ensures that the Scrum Team is provided with an environment conducive to completing the product's development 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.

Note that the Scrum Master role is very different from the role played by the Project Manager in a traditional Waterfall model of project management, in which the Project Manager works as a manager or leader for the project. The Scrum Master only works as a facilitator and he or she is at the same hierarchical level as anyone else in the Scrum Team—any person from the Scrum Team who learns how to facilitate Scrum projects can become the Scrum Master for a project or for a Sprint.

Corresponding to a Scrum Master role in a project, there could be a Program Scrum Master for a program or a Portfolio Scrum Master for a portfolio.

3. Scrum Team

The Scrum Team is a group or team of people who are responsible for understanding the business requirements specified by the Product Owner, estimating User Stories, and final creation of the project Deliverables.

Figure 3-1 presents an overview of the Core Scrum Team roles.

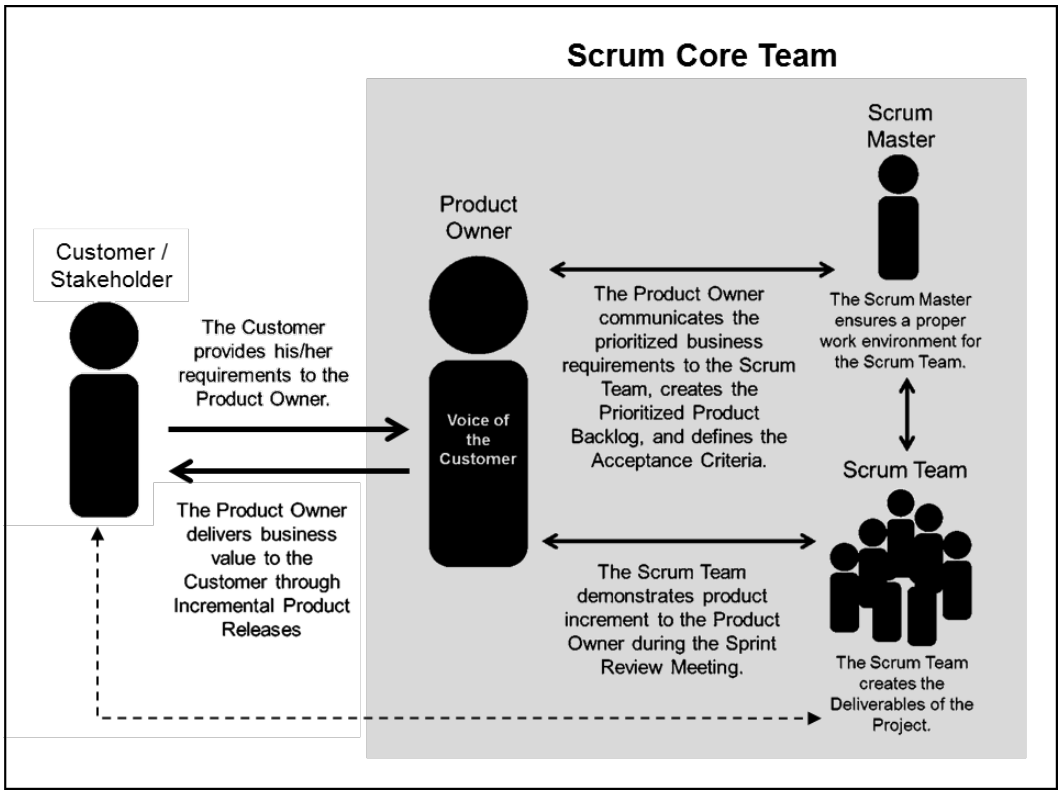


Figure 3-1: Scrum Roles—Overview

3.3.2 Non-core Roles

The non-core roles are those roles which are not mandatorily required for the Scrum project and may not be continuously or directly involved in the Scrum process. However, knowing non-core roles is important as they could play a significant part in some Scrum projects.

Non-core roles can include the following:

1. Stakeholder(s)

   Stakeholder(s) is a collective term that include customers, users, and sponsors, who frequently interface with the Product Owner, Scrum Master and Scrum Team to provide them with inputs and facilitate creation of the project's product, service, or other result. Stakeholder(s) influence the project throughout the project's development. Stakeholders may also have a role to play during the *Develop Epic(s), Create Prioritized Product Backlog, Conduct Release Planning, Retrospect Sprint*, and other important processes in Scrum.

- Customer

  The customer is the individual or the organization that acquires the project's product, service, or other result. For any organization, depending on the project, there can be both internal customers (i.e., within the same organization) or external customers (i.e., outside of the organization).

- Users

  Users are the individual or the organization that directly uses the project's product, service, or other result. Like customers, for any organization, there can be both internal and external users. Also, in some industries customers and users may be the same.

- Sponsor

  The sponsor is the individual or the organization that provides resources and support for the project. The sponsor is also the stakeholder to whom everyone is accountable in the end.

At times, the same person or organization can play multiple stakeholder roles; for example, the sponsor and the customer may be the same.

2. Vendors

Vendors include external individuals or organizations that provide products and services that are not within the core competencies of the project organization.

3. Scrum Guidance Body

The Scrum Guidance Body (SGB) is an optional role but highly recommended to formalize organizational practices related to Scrum. It generally consists of a group 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. These objectives guide the work carried out by the Product Owner, Scrum Master, and Scrum Team. The Scrum Guidance Body also helps capture the best practices that should be used across all Scrum projects in the organization.

The Scrum Guidance Body does not make decisions related to the project. Instead it acts as a consulting or guidance structure for all the hierarchy levels in the project organization—the portfolio, program, and project. Scrum Teams have the option of asking the Scrum Guidance Body for advice as required.

3.4 Product Owner

The Product Owner represents the interests of the stakeholder community to the Scrum Team. The Product Owner is responsible for ensuring clear communication of product or service functionality requirements to the Scrum Team, defining Acceptance Criteria, and ensuring those criteria are met. In other words, the Product Owner is responsible for ensuring that the Scrum Team delivers value. The Product Owner must always maintain a dual view. He or she must understand and support the needs and interests of all stakeholders, while also understanding the needs and workings of the Scrum Team. Because the Product Owner must understand the needs and priorities of the stakeholders, including customers and users, this role is commonly referred to as the Voice of the Customer.

Table 3-1 summarizes the Product Owner's responsibilities in the various Scrum processes.

Process	Product Owner Responsibilities
8.1 Create Project Vision	• Defines the Project Vision • Helps create the Project Charter and Project Budget
8.2 Identify Scrum Master and Stakeholder(s)	• Helps finalize Scrum Master for the project • Identifies Stakeholder(s)
8.3 Form Scrum Team	• Helps determine Scrum Team members • Helps develop a Collaboration Plan • Helps develop the Team Building Plan with Scrum Master(s)
8.4 Develop Epic(s)	• Creates Epic(s) and Personas
8.5 Create Prioritized Product Backlog	• Prioritizes Prioritized Product Backlog Items • Defines Done Criteria
8.6 Conduct Release Planning	• Creates Release Planning Schedule • Helps determine Length of Sprint
9.1 Create User Stories	• Helps create User Stories • Defines Acceptance Criteria for every User Story
9.2 Estimate User Stories	• Clarifies User Stories
9.3 Commit User Stories	• Works with Scrum Team to commit User Stories
9.4 Identify Tasks	• Explains User Stories to the Scrum Team while creating the Task List
9.5 Estimate Tasks	• Provides guidance and clarification to the Scrum Team in estimating effort for tasks
9.6 Create Sprint Backlog	• Clarifies requirements to the Scrum Team while creating the Sprint Backlog
10.1 Create Deliverables	• Clarifies business requirements to the Scrum Team
10.3 Groom Prioritized Product Backlog	• Grooms the Prioritized Product Backlog
11.1 Demonstrate and Validate	• Accepts/Rejects Deliverables

Sprints	• Provides necessary feedback to Scrum Master and Scrum Teams • Updates Release Plan and Prioritized Product Backlog
12.1 Ship Deliverables	• Helps deploy Product Releases and coordinates this with the customer
12.2 Retrospect Project	• Participates in Retrospective Sprint Meetings

Table 3-1: Responsibilities of the Product Owner in Scrum Processes

The other responsibilities of a Product Owner are:

- Determining the project's initial overall requirements and kicking off project activities; this may involve interaction with the Program Product Owner and the Portfolio Product Owner to ensure that the project aligns with direction provided by senior management.
- Representing user(s) of the product or service with a thorough understanding of the user community
- Securing the initial and ongoing financial resources for the project.
- Focusing on value creation and overall Return on Investment (ROI).
- Assessing the viability and ensuring the delivery of the product or service.

3.4.1 Voice of the Customer (VOC)

As the representative of the customer, the Product Owner is said to be the Voice of the Customer as he ensures that the explicit and implicit needs of the customer are translated into User Stories in the Prioritized Product Backlog and later on used to create project Deliverables for the customer.

3.4.2 Chief Product Owner

In the case of large projects with numerous Scrum Teams and multiple Product Owners, it is still necessary to have one single person who makes the day-to-day business decisions. This person/role is the Chief Product Owner. This role is responsible for coordinating the work of multiple Product Owners. With help from the Product Owners, the Chief Product Owner prepares and maintains the overall Prioritized Product Backlog for the large project, using it to coordinate work through the Product Owners of the Scrum Teams. The Chief Product Owner will be responsible for the final deliverable of the project whereas the Product Owners of the individual teams will be responsible only for those components and features being developed by their respective Scrum Teams.

In a large project, the Chief Product Owner will be tasked with prioritizing competing requests raised by the Product Owners based on their interaction with the stakeholders. The complexity of this task increases

greatly with each increase in the number of Scrum Teams and the number of Product Owners. An important part of the complexity of this task is to ensure various components are properly integrated and at appropriate times. Therefore, it is imperative to develop a list of components and resources needed in common for all teams throughout the project. Although the Chief Product Owner makes the final business decisions, he or she collaborates with the Chief Scrum Master, other Product Owners, and Scrum Masters to develop this list.

The Chief Product Owner also interfaces with the Program Product Owner to ensure alignment of the large project with the goals and objectives of the program.

Chief Product Owners should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Product Owner as well as Chapter 13, which describes Scaling Scrum for Large Projects.

3.4.3 Program Product Owner

The Program Product Owner defines the strategic objectives and priorities for the program. He or she is responsible for maximizing business value for a program by clearly articulating customer requirements and maintaining business justification for the program. The Program Product Owner also manages the Program Product Backlog.

The Program Product Owner interfaces with the Portfolio Product Owner to ensure alignment of the program with the goals and objectives of the portfolio. He or she is also involved with appointing Product Owners for individual projects, and ensuring that the vision, objectives, outcomes, and releases of the individual projects align with that of the program.

This role is similar to that of the Product Owner except it meets the needs of the program or business unit rather than of a single Scrum Team.

Program Product Owners should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Product Owner as well as Chapter 14, which describes Scaling Scrum for the Enterprise.

3.4.4 Portfolio Product Owner

The Portfolio Product Owner makes the decisions at the portfolio level. He or she will have the best perspective to help decide how to organize the enterprise to meet the vision. The role of the Portfolio Product Owner is similar to that of the Program Product Owner in a program. He or she is responsible for and drives the creation and grooming of the Portfolio Product Backlog.

This role is similar to that of the Product Owner except it meets the needs of the portfolio or business unit rather than of a single Scrum Team.

Portfolio Product Owners should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Product Owner as well as Chapter 14, which describes Scaling Scrum for the Enterprise.

3.5 Scrum Master

The Scrum Master is the "servant leader" of the Scrum Team who moderates and facilitates team interactions as team coach and motivator. The Scrum Master is responsible for ensuring that the team has a productive work environment by guarding the team from external influences, removing any obstacles, and enforcing Scrum principles, aspects, and processes.

Table 3-2 summarizes the Scrum Master's responsibilities in the various Scrum processes.

Processes	Scrum Master Responsibilities
8.2 Identify Scrum Master and Stakeholder(s)	• Helps identify Stakeholder(s) for the project
8.3 Form Scrum Team	• Facilitates selection of the Scrum Team • Facilitates creation of the Collaboration Plan and the Team Building Plan • Ensures back-up resources are available for smooth project functioning
8.4 Develop Epic(s)	• Facilitates creation of Epic(s) and Personas
8.5 Create Prioritized Product Backlog	• Helps Product Owner in creation of the Prioritized Product Backlog and in definition of the Done Criteria
8.6 Conduct Release Planning	• Coordinates creation of Release Planning Schedule • Determines Length of Sprint
9.1 Create User Stories	• Assists the Scrum Team in creating User Stories and their Acceptance Criteria
9.2 Estimate User Stories	• Facilitates meetings of the Scrum Team to estimate User Stories
9.3 Commit User Stories	• Facilitates meetings of the Scrum Team to commit User Stories
9.4 Identify Tasks	• Facilitates the Scrum Team in creating the Task List for the next Sprint
9.5 Estimate Tasks	• Assists the Scrum Team in estimating the effort required to complete the tasks agreed to for the Sprint
9.6 Create Sprint Backlog	• Assists the Scrum Team in developing the Sprint Backlog and the Sprint Burndown Chart
10.1 Create Deliverables	• Supports the Scrum Team in creating the Deliverables agreed to for the Sprint • Helps update the Scrumboard and the Impediment Log
10.2 Conduct Daily Standup	• Ensures that the Scrumboard and the Impediment Log remain updated

10.3 Groom Prioritized Product Backlog	• Facilitates Prioritized Product Backlog Review Meetings
11.1 Demonstrate and Validate Sprints	• Facilitates presentation of completed Deliverables by the Scrum Team for the Product Owner's approval
11.2 Retrospect Sprint	• Ensures that ideal project environment exists for the Scrum Team in the succeeding Sprints
12.2 Retrospect Project	• Represents the Scrum Core Team to provide lessons from the current project, if necessary

Table 3-2: Responsibilities of the Scrum Master in Scrum Processes

3.5.1 Chief Scrum Master

Large projects require multiple Scrum Teams to work in parallel. Information gathered from one team may need to be appropriately communicated to other teams—the Chief Scrum Master is responsible for this activity.

The role of a Chief Scrum Master is necessary to ensure proper collaboration among the Scrum Teams. Coordination across various Scrum Teams working on a project is typically done through the Scrum of Scrums (SoS) Meeting (see section 13.2.2.1). There is no hierarchy between the Scrum Masters: they are all peers. The Chief Scrum Master just works on a multi-team level, whereas the Scrum Masters work on a single team level.

Figure 3-2 provides questions that are asked during a Scrum of Scrums (SoS) Meeting.

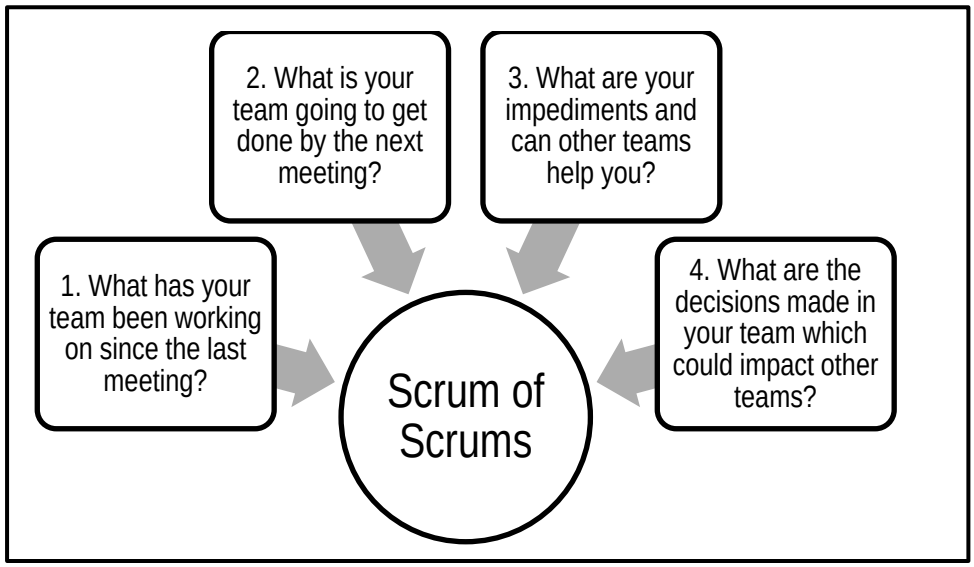


Figure 3-2: Questions asked during a Scrum of Scrums Meeting

Typically, any inter-team issues are addressed by the interested parties in a session immediately following the Scrum of Scrums Meeting. The Chief Scrum Master facilitates this session.

The Chief Scrum Master can be chosen from the Scrum Masters of the large project or can be someone else. For very large projects, it is recommended to have a Chief Scrum Master who is not also a Scrum Master for an individual project because the effort required for the Chief Scrum Master role will prevent the Chief Scrum Master from also being able to dedicate enough time to the work with his/her Scrum Team. In either case, the Chief Scrum Master should have enough Scrum expertise to be able to foster collaboration and to help and coach others with the implementation of Scrum for a smooth delivery of the project's products.

Apart from clearing impediments and ensuring a conducive project environment for the Scrum Teams, the Chief Scrum Master also collaborates with the Chief Product Owner, other Scrum Masters, and Product Owners in activities such as developing the list of components and resources needed in common for all teams throughout the project. He or she facilitates everything that goes beyond the realm of a single Scrum Team.

The Chief Scrum Master also interfaces with the Program Scrum Master to ensure alignment of the large project with the goals and objectives of the program.

Chief Scrum Masters should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Scrum Master as well as Chapter 13, which describes Scaling Scrum for Large Projects.

3.5.2 Program Scrum Master

The Program Scrum Master is a facilitator who ensures that all project teams in the program are provided with an environment conducive to completing their projects successfully. The Program Scrum Master guides, facilitates, and teaches Scrum practices to everyone involved in the program; provides guidance to Scrum Masters of individual projects; clears impediments for the different project teams; coordinates with the Scrum Guidance Body to define objectives related to quality, government regulations, security, and other key organizational parameters; and ensures that Scrum processes are being effectively followed throughout the program.

The Program Scrum Master interfaces with the Portfolio Scrum Master to ensure alignment of the program with the goals and objectives of the portfolio. He or she is also involved with appointing Scrum Masters for individual projects and ensuring that the vision, objectives, outcomes, and releases of individual projects in the program align with that of the program.

This role is similar to that of the Scrum Master except it meets the needs of the program or business unit rather than of a single Scrum Team.

Program Scrum Masters should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Scrum Master as well as Chapter 14, which describes Scaling Scrum for the Enterprise.

3.5.3 Portfolio Scrum Master

This role is similar to that of the Scrum Master except it meets the needs of the portfolio or business unit rather than of a single Scrum Team.

Portfolio Scrum Masters should refer to the *SBOK™ Guide* sections defined in the Roles Guide for the Scrum Master as well as Chapter 14, which describes Scaling Scrum for the Enterprise.

3.6 Scrum Team

The Scrum Team is sometimes referred to as the Development Team since they are responsible for developing the product, service, or other result. It consists of a group of individuals who work on the User Stories in the Sprint Backlog to create the Deliverables for the project.

Table 3-3 summarizes the Scrum Team's responsibilities in the various Scrum processes.

Processes	Scrum Team Responsibilities
8.3 Form Scrum Team	• Provides inputs for creation of the Collaboration Plan and the Team Building Plan
8.4 Develop Epic(s)	• Ensures a clear understanding of Epic(s) and Personas
8.5 Prioritized Product Backlog	• Understands the User Stories in the Prioritized Product Backlog
8.6 Conduct Release Planning	• Agrees with other Scrum Core Team members on the Length of Sprint • Seeks clarification on new products or changes in the existing products, if any, in the refined Prioritized Product Backlog
9.1 Create User Stories	• Provides inputs to the Product Owner on creation of User Stories
9.2 Estimate User Stories	• Estimates User Stories approved by the Product Owner
9.3 Commit User Stories	• Commits User Stories to be done in a Sprint
9.4 Identify Tasks	• Develops Task List based on agreed User Stories and dependencies
9.5 Estimate Tasks	• Estimates effort for tasks identified and if necessary, updates the Task List
9.6 Create Sprint Backlog	• Develops the Sprint Backlog and the Sprint Burndown Chart
10.1 Create Deliverables	• Creates Deliverables • Identifies risks and implements risk mitigation actions, if any • Updates Impediment Log and dependencies
10.2 Conduct Daily Standup	• Updates Burndown Chart, Scrumboard, and Impediment Log

	• Discusses issues faced by individual members and seeks solutions to motivate the team • Identifies risks, if any • Submits Change Requests, if required
10.3 Groom Prioritized Product Backlog	• Participates in Prioritized Product Backlog Review Meetings
11.1 Demonstrate and Validate Sprints	• Demonstrates completed deliverables to the Product Owner for approval
11.2 Retrospect Sprint	• Identifies improvement opportunities, if any, from the current Sprint and agrees on any actionable improvements for the next Sprint
12.2 Retrospect Project	• Participates in the Retrospect Project Meeting

Table 3-3: Responsibilities of the Scrum Team in Scrum Processes

3.6.1 Personnel Selection

Figure 3-3 lists the desirable traits for the core Scrum roles.

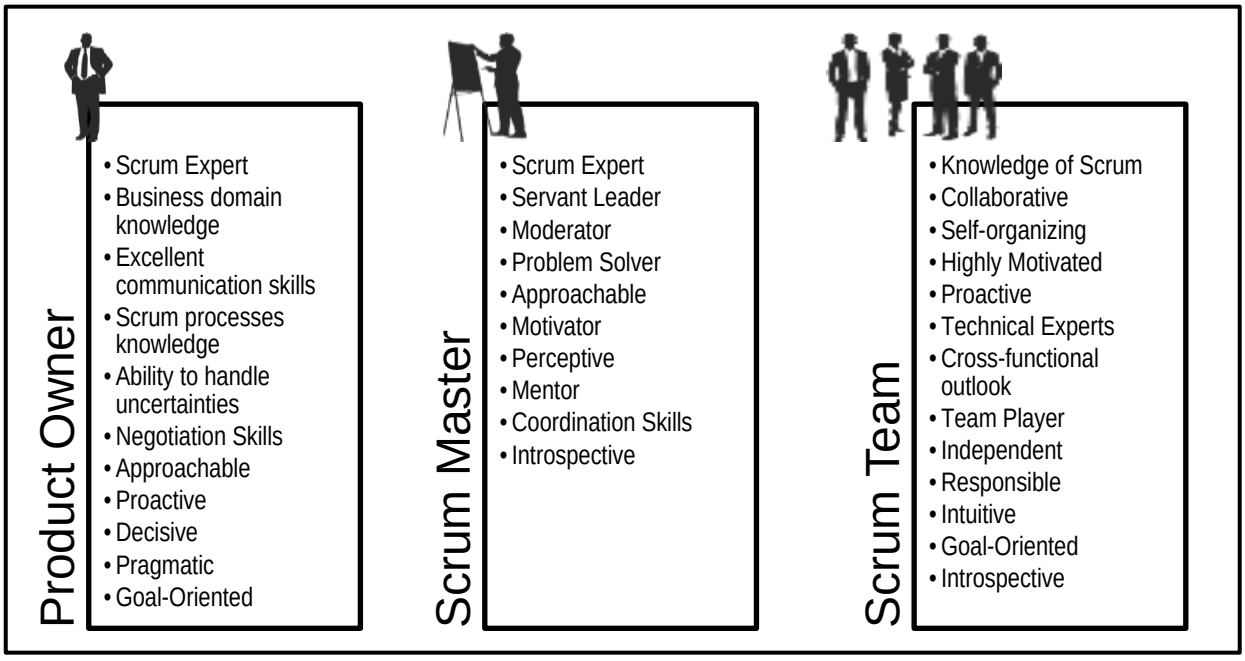


Figure 3-3: Desirable Traits for the Core Scrum Roles

3.6.2 Scrum Team Size

It is important for the Scrum Team to possess all the essential skills required to carry out the work of the project. It is also necessary to have a high level of collaboration to maximize productivity, so that minimal coordination is required to get things done.

The optimum size for a Scrum Team is six to ten members—large enough to ensure adequate skill sets, but small enough to collaborate easily. A key benefit of a six-to-ten-member team is that communication and management are typically simple and require minimal effort. However, there may also be drawbacks. One major drawback is that smaller teams are more significantly impacted by the loss of a team member than larger teams, even for a short period of time. To address this problem, it may be possible for team members to have expert knowledge and skills outside their own specific role. However, this may be difficult and depends on the type of project, industry, and size of the organization. It is also recommended to have back-up persons to replace any person who may have to leave the Scrum Team.

3.7 Scrum in Projects, Programs, and Portfolios

3.7.1 Definition of Project, Program, and Portfolio

- Project—A project is a collaborative enterprise to either create new products or services or to deliver results as defined in the Project Vision Statement. Projects are usually impacted by constraints of time, cost, scope, quality, people and organizational capabilities. The objective of the project team is to create Deliverables as defined in Prioritized Product Backlog.
- Program—A program is a group of related projects, with the objective to deliver business outcomes as defined in the Program Vision Statement. The Prioritized Program Backlog incorporates the Prioritized Product Backlogs for all the projects in the program.
- Portfolio—A portfolio is a group of related programs, with the objective to deliver business outcomes as defined in the Portfolio Vision Statement. The Prioritized Portfolio Backlog incorporates the Prioritized Program Backlogs for all the programs in the portfolio.

The following are examples of projects, programs, and portfolios from different industries and sectors:

Example 1: Construction Company

- Project—Construction of a house
- Program—Construction of a housing complex
- Portfolio—All the housing projects of the company

Example 2: Aerospace Organization

- Project—Building the launch vehicle
- Program—Successful launch of a satellite
- Portfolio—All the active satellite programs

Example 3: Information Technology (IT) Company

- Project—Development of the shopping cart module
- Program—Development of a fully functional e-commerce website
- Portfolio—All the websites developed by the company so far

Figure 3-4 illustrates how Scrum can be used across the organization for portfolios, programs, or projects.

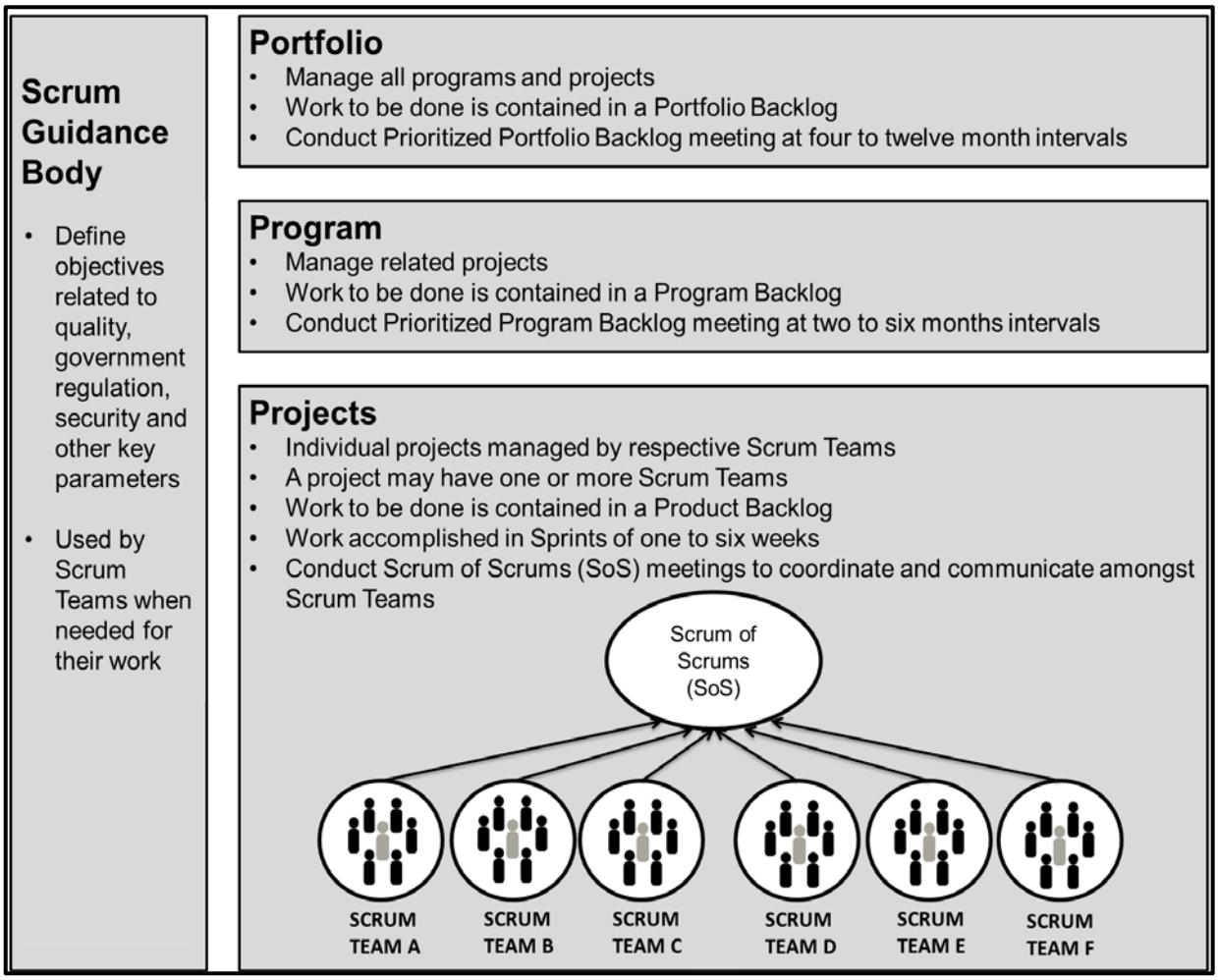


Figure 3-4: Scrum Across the Organization for Projects, Programs, and Portfolios

3.7.1.1 Working with Portfolio and Program Teams

When applying Scrum to manage projects within the context of a program or portfolio, it is strongly recommended that the general principles of Scrum presented in this publication are adhered to. It is understood though, that in order to accommodate the overall program or portfolio activities and interdependencies, minor adjustments to the set of tools, as well as the organizational structure may be required. If the Scrum Guidance Body exists, it may be responsible to scrutinize the organization at different levels to understand and define appropriate application of Scrum, and to act as a consulting body for everyone working on a project, program, or portfolio.

Portfolios and programs have separate teams with different sets of objectives. Program management teams aim to deliver capabilities and realize certain goals that contribute toward the achievement of specific program objectives. In contrast, the portfolio team has to balance the objectives of various programs to achieve the strategic objectives of the organization as a whole. Chapter 14 goes into detail on Scaling Scrum for the Enterprise.

3.7.1.2 Managing Communication with Portfolio and Program Teams

The problems and issues faced when using Scrum within a program or portfolio primarily involve coordination across numerous teams. This can lead to failure if not carefully managed. Tools used for communication need to be scaled to match the requirements of the many teams involved in a program or portfolio. Each Scrum Team must address not only internal communications, but also external communications with other teams and the relevant stakeholders of the program or portfolio.

3.7.2 Maintaining Stakeholder Involvement

Scrum requires complete support from the project stakeholders. The responsibility for keeping stakeholders engaged lies with the Product Owner. The following are actions recommended for maintaining stakeholder engagement and support:

- Ensure effective collaboration and stakeholder involvement in the project
- Continually assess business impact
- Maintain regular communication with stakeholders
- Manage stakeholders' expectations

One key stakeholder is the sponsor—the individual who provides the funding and other resources for a project. Sponsors want to understand the financial bottom line related to a product or service and are typically more concerned with final outcomes rather than with individual tasks.

It is important that the sponsors who are funding the project have clarity on the following issues:

- Benefits of implementing Scrum
- Target deadlines and estimated costs of Scrum projects
- Overall risks involved in Scrum projects and the steps to mitigate them
- Expected release dates and final Deliverables

3.8 Summary of Responsibilities

Role	Responsibilities
Scrum Team	• Takes collective responsibility and ensures that the project deliverables are created per requirements • Assures Product Owner and Scrum Master that the allocated work is being performed according to plan
Product Owner/ Chief Product Owner	• Creates the project's initial overall requirements and gets the project rolling • Helps appoint appropriate people to the Scrum Master and Scrum Team roles • Helps secure the initial and ongoing financial resources for the project • Determines Product Vision • Assesses the viability and ensures delivery of the product or service • Ensures transparency and clarity of Prioritized Product Backlog Items • Decides minimum marketable release content • Provides Acceptance Criteria for the User Stories to be developed in a Sprint • Inspects deliverables • Decides Sprint duration
Scrum Master/ Chief Scrum Master	• Ensures that Scrum processes are correctly followed by all team members including the Product Owner • Ensures that development of the product or service is progressing smoothly and the Scrum Team members have all the necessary tools to get the work done • Oversees Release Planning Meeting and schedules other meetings
Program Product Owner	• Defines the strategic objectives and priorities for programs
Program Scrum Master	• Solves problems and coordinates meetings for programs
Portfolio Product Owner	• Defines the strategic objectives and priorities for portfolios
Portfolio Scrum Master	• Solves problems and coordinates meetings for portfolios
Stakeholder(s)	• Is a collective term that includes customers, users, and sponsors • Frequently interface with the Product Owner, Scrum Master, and Scrum Team to provide them inputs and facilitate creation of the Deliverables of the project.
Scrum Guidance Body	• Establishes overall guidelines and metrics for developing role descriptions for Scrum Team members • Acts as a consultant to projects across organization at different levels • Understands and defines appropriate levels of grouping, roles, and meetings for Scrum projects

Table 3-4: Summary of Responsibilities Relevant to Organization

3.9 Scrum vs. Traditional Project Management

Organization structure and definition of roles and associated responsibilities are some of the areas where Scrum differs in a major way from traditional project management methods.

In traditional project management methods, the organization structure is hierarchical and authority for all aspects of the project is delegated from higher level to lower (e.g., project sponsor delegates authority to project manager and the project manager delegates authority to team members). Traditional project management methods emphasize on individual accountability for project responsibilities rather than group ownership or accountability. Any deviation from the delegated authority is looked at as a sign of issues and may be escalated to the higher level in the organization hierarchy. It is usually the project manager who is responsible for successful completion of the project and he or she takes decisions on various aspects of the project, including initiating, planning, estimating, executing, monitoring and controlling, and closing.

The emphasis in Scrum is on self-organization and self-motivation where the team assumes greater responsibility in making a project successful. This also ensures that there is team buy-in and shared ownership. This, in turn, results in team motivation leading to an optimization of team efficiencies. The Product Owner, Scrum Master, and the Scrum Team work very closely with relevant Stakeholder(s) for refining requirements as they go through the *Develop Epic(s)*, *Create Prioritized Product Backlog,* and *Create User Stories* processes. This ensures that there is no scope for isolated planning in Scrum. Team experience and expertise in product development are used to assess the inputs needed to plan, estimate and execute project work. Collaboration among Scrum Core Team members ensures that the project is carried out in an innovative and creative environment that is conducive to growth and team harmony.

3.10 Popular HR Theories and their Relevance to Scrum

3.10.1 Tuckman's Model of Group Dynamics

The Scrum approach and method may initially seem quite different and difficult for a new Scrum Team. A new Scrum Team, like any other new team, generally evolves through a four-stage process during its first Scrum project. This process is known as Tuckman's Model of group dynamics (Tuckman, 1965). The main idea is that the four stages—Forming, Storming, Norming and Performing—are imperative for a team to develop by mitigating problems and challenges, finding solutions, planning work, and delivering results.

The four stages of the model are the following:

1. Forming—This is often experienced as a fun stage because everything is new and the team has not yet encountered any difficulties with the project.
2. Storming—During this stage, the team tries to accomplish the work; however, power struggles may occur, and there is often chaos or confusion among team members.
3. Norming—This is when the team begins to mature, sort out their internal differences, and find solutions to work together. It is considered a period of adjustment.
4. Performing—During this stage, the team becomes its most cohesive, and it operates at its highest level in terms of performance. The members have evolved into an efficient team of peer professionals who are consistently productive.

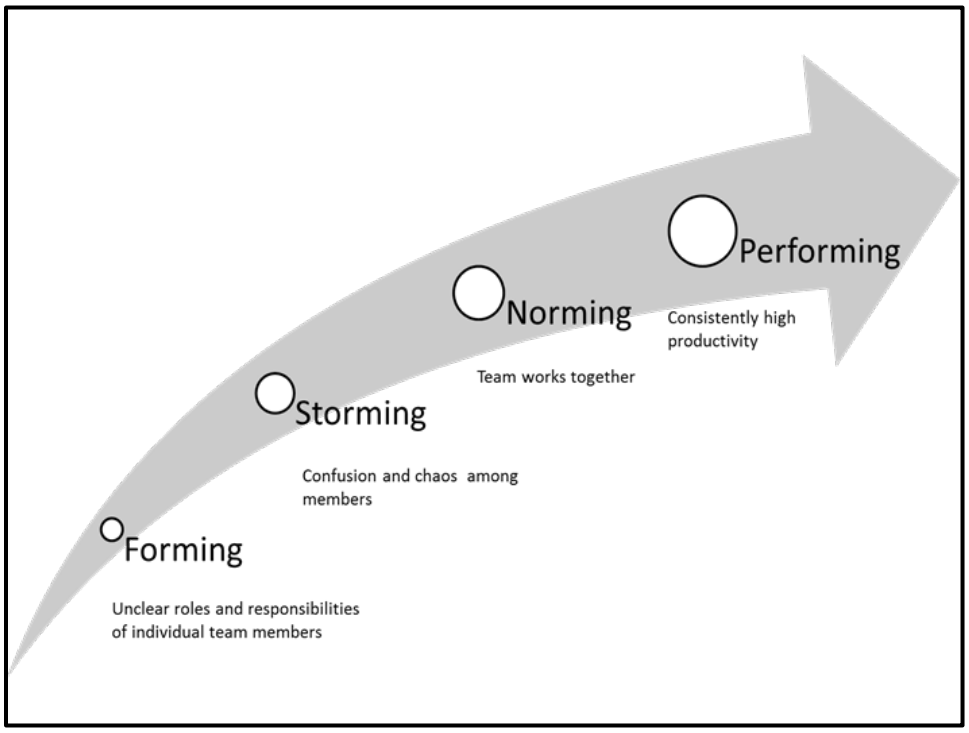


Figure 3-5: Tuckman's Stages of Group Development

3.10.2 Conflict Management

Organizations applying the Scrum framework encourage an open environment and dialogue among employees. Conflicts among Scrum Team members are generally resolved independently, with little or no involvement from management or others outside the Scrum Team.

Conflict can be healthy when it promotes team discussions and encourages debates, as this usually results in benefits for the project and the respective team members. It is therefore important that the resolution of conflicts be encouraged, promoting an open environment where team members feel welcome to express their opinions and concerns with each other and about the project, and ultimately agree on what is to be delivered and how the work in each Sprint will be performed.

Conflict management techniques are used by team members to manage any conflicts that arise during a Scrum project. Sources of conflict evolve primarily due to schedules, priorities, resources, reporting hierarchy, technical issues, procedures, personality, and costs.

3.10.3 Conflict Management Techniques

Usually there are four approaches to managing conflict in an organization applying Scrum processes:

1. Win-Win
2. Lose-Win
3. Lose-Lose
4. Win-Lose

3.10.3.1 Win-Win

It is usually best for team members to face problems directly with a cooperative attitude and an open dialogue to work through any disagreements to reach consensus. This approach is called *Win-Win*. Organizations implementing Scrum should promote an environment where employees feel comfortable to openly discuss and confront problems or issues and work through them to reach Win-Win outcomes.

3.10.3.2 Lose-Win

Some team members may at times feel that their contributions are not being recognized or valued by others, or that they are not being treated equally. This may lead them to withdraw from contributing effectively to the project and agree to whatever they are being told to do, even if they are in disagreement. This approach is called Lose-Win. This situation may happen if there are members in the team (including managers) who use an authoritative or directive style of issuing orders and/or do not treat all team members equally. This

approach is not a desired conflict management technique for Scrum projects, since active contribution of every member of the team is mandatory for successful completion of each Sprint. The Scrum Master should encourage the involvement of any team members who appear to be withdrawing from conflict situations. For example, it is important for all team members to speak and contribute at each Daily Standup Meeting so that any issues or impediments can be made known and managed effectively.

3.10.3.3 Lose-Lose

In conflict situations, team members may attempt to bargain or search for solutions that bring only a partial degree or temporary measure of satisfaction to the parties in a dispute. This situation could happen in Scrum Teams where team members try to negotiate for suboptimal solutions to a problem. This approach typically involves some "give and take" to satisfy every team member—instead of trying to solve the actual problem. This generally results in an overall *Lose-Lose* outcome for the individuals involved and consequently the project. The Scrum Team should be careful to ensure that team members do not get into a Lose-Lose mentality. Scrum Daily Standup and other Scrum meetings are conducted to ensure that actual problems get solved through mutual discussions.

3.10.3.4 Win-Lose

At times, a Scrum Master or another influential team member may believe he or she is a de facto leader or manager and try to exert their viewpoint at the expense of the viewpoints of others. This conflict management technique is often characterized by competitiveness and typically results in a *Win-Lose* outcome. This approach is not recommended when working on Scrum projects, because Scrum Teams are by nature self-organized and empowered, with no one person having true authority over another team member. Although the Scrum Team may include persons with different levels of experience and expertise, every member is treated equally and no person has the authority to be the primary decision maker.

3.10.4 Leadership Styles

Leadership styles vary depending on the organization, the situation, and even the specific individuals and objectives of the Scrum project. Some common leadership styles are as follows:

- Servant Leadership—Servant leaders employ listening, empathy, commitment, and insight while sharing power and authority with team members. Servant leaders are stewards who achieve results by focusing on the needs of the team. This style is the embodiment of the Scrum Master role.
- Delegating—Delegating leaders are involved in the majority of decision making; however, they delegate some planning and decision-making responsibilities to team members, particularly if they are competent

to handle the assigned tasks. This leadership style is appropriate in situations where the leader is in tune with specific project details, and when time is limited.

- Autocratic—Autocratic leaders make decisions on their own, allowing team members little, if any involvement or discussion before a decision is made. This leadership style should only be used on rare occasions.
- Directing— Directing leaders instruct team members which tasks are required, when they should be performed and how they should be performed.
- Laissez Faire—With this leadership style, the team is left largely unsupervised, so the leader does not interfere with their daily work activities. Often this style leads to a state of anarchy.
- Coaching/Supportive—Coaching and supportive leaders issue instructions and then support and monitor team members through listening, assisting, encouraging, and presenting a positive outlook during times of uncertainty.
- Task-Oriented—Task-oriented leaders enforce task completion and adherence to deadlines.
- Assertive—Assertive leaders confront issues and display confidence to establish authority with respect.

3.10.4.1 Servant Leadership

The preferred leadership style for Scrum projects is Servant Leadership. This term was first described by Robert K. Greenleaf in an essay entitled *The Servant as Leader*. Below is an excerpt where he explains the concept:

> The servant-leader *is* servant first…It begins with the natural feeling that one wants to serve, to serve *first*. Then conscious choice brings one to aspire to lead. That person is sharply different from one who is *leader* first, perhaps because of the need to assuage an unusual power drive or to acquire material possessions…The leader-first and the servant-first are two extreme types. Between them there are shadings and blends that are part of the infinite variety of human nature….
>
> The difference manifests itself in the care taken by the servant-first to make sure that other people's highest priority needs are being served. The best test, and difficult to administer, is: Do those served grow as persons? Do they, *while being served*, become healthier, wiser, freer, more autonomous, more likely themselves to become servants? *And*, what is the effect on the least privileged in society? Will they benefit or at least not be further deprived? (Greenleaf 1970, 6)

Elaborating on the writings of Greenleaf, Larry Spears identifies ten traits that every effective servant-leader should possess:

1. Listening—Servant leaders are expected to listen intently and receptively to what is being said, or not said. They are able to get in touch with their inner voice to understand and reflect on their own feelings.

2. Empathy—Good servant leaders accept and recognize individuals for their special and unique skills and abilities. They assume workers have good intentions and accept them as individuals, even when there are behavioral or performance issues.

3. Healing—The motivation and potential to heal oneself and one's relationship with others is a strong trait of servant leaders. Servant leaders recognize and take the opportunity to help their colleagues who are experiencing emotional pain.

4. Awareness—Awareness and particularly self-awareness is a trait of servant leaders. This allows them to better understand and integrate issues such as those related to ethics, power, and values.

5. Persuasion—Servant leaders use persuasion, rather than their positional authority to gain group consensus and make decisions. Rather than forcing compliance and coercion as is typical in some authoritarian management styles, servant leaders practice persuasion.

6. Conceptualization—The ability to view and analyze problems (in an organization) from a broader conceptual and visionary perspective, rather than focusing on merely the immediate short-term goals, is a unique skill of good servant leaders.

7. Foresight—Their intuitive minds allow servant leaders to use and apply past lessons and present realities to foresee the outcome of current situations and decisions.

8. Stewardship—Stewardship demands a commitment to serving others. Servant leaders prefer persuasion over control to ensure that they gain the trust of others in the organization.

9. Commitment to the growth of others—Servant leaders have a deep commitment to the growth of people within their organization. They take on the responsibility of nurturing the personal, professional, and spiritual growth of others (e.g., providing access to resources for personal and professional development, encouraging workers to participate in decision making).

10. Building community—Servant leaders are interested in building communities within a working environment, particularly given the shift in societies away from smaller communities to large institutions shaping and controlling human lives.

Scrum believes that all leaders of Scrum projects (including the Scrum Master and Product Owner) should be servant-leaders who have the above traits.

3.10.5 Maslow's Hierarchy of Needs Theory

Maslow (1943) presented a need hierarchy which recognizes that different people are at different levels in their needs. Usually people start out looking for physiological needs and then progressively move up the needs hierarchy.

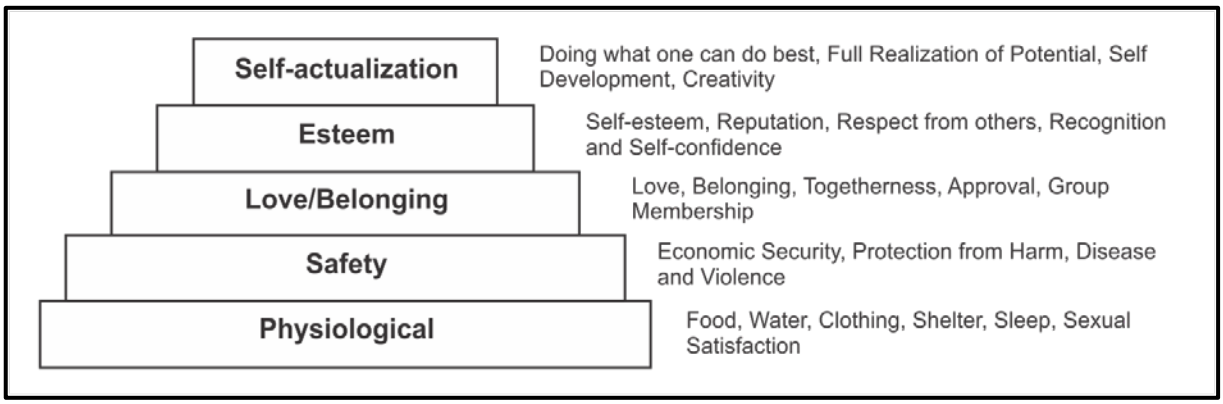


Figure 3-6: Maslow's Hierarchy of Needs Theory

To be successful, a Scrum Team needs both core and non-core team members who have reached the esteem or self-actualization levels. The concept of self-organizing teams, which is a key principle in Scrum, requires team members to be self-motivated, and to participate and contribute fully towards meeting the project goals.

As a leader, the Scrum Master needs to understand where each person on the team is relative to the pyramid. This understanding will help to determine the best approach in motivating each individual.

Additionally, everyone fluctuates up and down the levels in the needs hierarchy throughout life due to their own motivation and efforts to move up the hierarchy or sometimes due to factors beyond their control that may push them down. The Scrum Master's goal is to work with individuals on the team to build their skills and knowledge and help them move up the needs hierarchy. This support results in a team that consists of individuals who are motivated and strong contributors to the project and to the organization as a whole.

3.10.6 Theory X and Theory Y

Douglas McGregor (1960) proposed two management theories:

- Theory X—Theory X leaders assume employees are inherently unmotivated and will avoid work if possible, warranting an authoritarian style of management.

- Theory Y—Theory Y leaders, on the other hand, assume employees are self-motivated and seek to accept greater responsibility. Theory Y involves a more participative management style.

Scrum projects are not likely to be successful with organizations that have Theory X leaders in the roles of Scrum Master or Product Owner. All leaders in Scrum projects should subscribe to Theory Y, whereby they view individuals as important assets and work towards developing their team members' skills and empowering their team members while expressing appreciation for the work the team has completed to accomplish the project objectives.

4. BUSINESS JUSTIFICATION

4.1 Introduction

The purpose of this chapter is to understand the concept and purpose of Business Justification as it relates to Scrum projects. It is important for an organization to perform a proper business justification and create a viable Project Vision Statement prior to starting any project. This helps key decision makers understand the business need for a change or for a new product or service and the justification for moving forward with a project. It also helps the Product Owner to create a Prioritized Product Backlog along with the business expectations of Senior Management & Stakeholder(s).

Business Justification, as defined in *A Guide to the Scrum Body of Knowledge* (*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 deliverables. 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 chapter is divided into the following sections:

4.2 Roles Guide—This section provides guidance on which sections are relevant for each of the core Scrum roles: Product Owner, Scrum Master, and Scrum Team.

4.3 Value-driven Delivery—This section describes the concept of business value and its importance in any project. It also provides information regarding the responsibilities of the various individuals including the Product Owner, involved in achieving business value.

4.4 Importance of Business Justification—This section details the importance of business justification, the factors that determine it, and how it is maintained and verified throughout the project.

4.5 Business Justification Techniques—This section describes in detail how business justification is assessed and verified using various tools.

4.6 Continuous Value Justification—This section details the importance of continuous value justification and expands on how it is achieved.

4.7 Confirm Benefits Realization—This section describes how benefits are realized throughout the project.

4.8 Summary of Responsibilities—This section defines the responsibilities relevant to business justification for project team members based on their roles.

4.9 Scrum vs. Traditional Project Management—This section highlights the business benefits of the Scrum method over traditional project management models.

4.2 Roles Guide

1. Product Owner—Business justification is primarily conducted by the Product Owner; therefore, this entire chapter is most applicable to this role.
2. Scrum Master—The Scrum Master should be familiar with this entire chapter, with primary focus on sections 4.3, 4.4, 4.6, 4.7 and 4.8.
3. Scrum Team—The Scrum Team should focus primarily on sections 4.3, 4.7 and 4.8.

4.3 Value-driven Delivery

A project is a collaborative enterprise to either create new products or services or to deliver results as defined in the Project Vision Statement. Projects are usually impacted by constraints of time, cost, scope, quality, people, and organizational capabilities. Usually, the results generated by projects are expected to create some form of business or service value.

Since value is a primary reason for any organization to move forward with a project, Value-driven Delivery must be the main focus. Delivering value is ingrained in the Scrum framework. Scrum facilitates delivery of value very early on in the project and continues to do so throughout the project lifecycle.

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, it is therefore important 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.

In order to provide Value-driven Delivery, it is important to:

1. Understand what adds value to customers and users and to prioritize the high value requirements on the top of the Prioritized Product Backlog.
2. Decrease uncertainty and constantly address risks that can potentially decrease value if they materialize. Also work closely with project stakeholders showing them product increments at the end of each Sprint, enabling effective management of changes.

3. *Create Deliverables* based on the priorities determined by producing potentially shippable product increments during each Sprint so that customers start realizing value early on in the project.

The concept of Value-driven Delivery in Scrum makes the Scrum framework very attractive for business stakeholders and senior management. This concept is very different when compared with traditional project management models where:

1. Requirements are not prioritized by business value.
2. Changing requirements after project initiation is difficult and can only be done through a time consuming change management process.
3. Value is realized only at the end of the project when the final product or service is delivered.

Figure 4-1 contrasts Value-driven Delivery in Scrum versus Traditional projects.

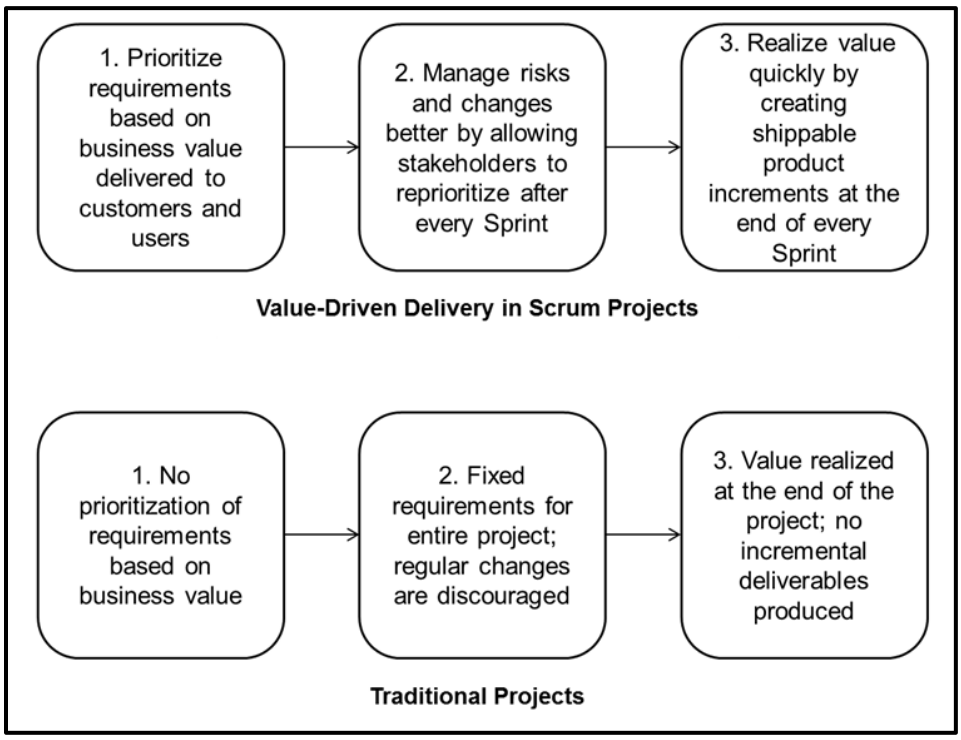


Figure 4-1: Delivering Value in Scrum vs. Traditional Projects

4.3.1 Responsibilities of the Product Owner in Business Justification

The responsibility of prioritizing and delivering business value in an organization for projects lies primarily with the Product Owner. For programs and portfolios, the responsibility lies with the Program Product Owner and Portfolio Product Owner, respectively. Their role is to act as effective representatives of the customer and/or sponsor. The guidelines for evaluating and measuring business value may typically be set forth by a Scrum Guidance Body.

Figure 4-2 illustrates the business justification responsibilities in a hierarchical order.

Portfolio Product Owner	• Delivers value for portfolios • Creates business justification for portfolios • Provides value guidance for programs • Approves business justification for programs
Program Product Owner	• Delivers value for programs • Creates business justification for programs • Provides value guidance for projects • Approves business justification for projects
Product Owner	• Delivers value for projects • Creates business justification for projects • Confirms benefit realization to stakeholders

Figure 4-2: Hierarchy for Business Justification Responsibilities

4.3.2 Responsibilities of Other Scrum Roles in Business Justification

It is important to note that although the Product Owner is primarily responsible for business justification, other persons working in Scrum projects also contribute significantly as follows:

1. The sponsor provides funding for the project and constantly monitors the project to confirm realization of benefits.

2. Customers and users are involved in defining the prioritized list of requirements and User Stories in the Prioritized Product Backlog, reviewing Deliverables after every Sprint or release, and confirming that benefits are realized.

3. The Scrum Guidance Body may provide guidelines and recommendations related to business justification techniques and confirming benefits realization and so forth. Such guidelines and recommendations may then be referred to by Scrum Core Teams and Stakeholder(s).

4. The Scrum Master facilitates creation of the project's deliverables; manages risks, changes, and impediments during *Conduct Daily Standup*, *Retrospect Sprint*, and other Scrum processes. The Scrum Master coordinates with the Scrum Team to create the deliverables and with the Product Owner and other stakeholders to ensure that benefits from the project are realized.

5. The Scrum Team works on creating the deliverables of the project and contributes to realizing business value for all stakeholders and the project. The Scrum Team is also involved in the *Develop Epic(s)*, *Create Prioritized Product Backlog*, *Create User Stories*, *Estimate User Stories, Commit User Stories*, and associated processes where the business requirements are defined and prioritized. The Scrum Team also helps in identifying risks and submits Change Requests for improvements in Sprint Retrospect Meetings and other meetings.

4.4 Importance of Business Justification

Business justification demonstrates the reasons for undertaking a project. It answers the question "Why is this project needed?" Business justification drives all decision making related to a project. So, it is important to assess the viability and achievability of a project not only before committing to significant expenditures or investment at initial stages of the project but also to verify the business justification for continuance throughout the project's lifecycle. A project should be terminated if it is found to be unviable; the decision should be escalated to the relevant stakeholders and to senior management. The business justification for a project must be assessed at the beginning of the project, at pre-defined intervals throughout the project, and at any time when major issues or risks that threaten the project viability arise.

4.4.1 Factors Used to Determine Business Justification

There are numerous factors a Product Owner must consider to determine the business justification for a project. The following are some of the most important factors:

1. Project Reasoning

   Project reasoning includes all factors which necessitate the project, whether positive or negative, chosen or not (e.g., inadequate capacity to meet existing and forecasted demand, decrease in customer satisfaction, low profit, legal requirement, etc.).

2. Business Needs

   Business needs are those business outcomes that the project is expected to fulfill, as documented in the Project Vision Statement.

3. Project Benefits

   Project benefits include all measurable improvements in a product, service, or result which could be provided through successful completion of a project.

4. Opportunity Cost

   Opportunity cost covers the next best business option or project that was discarded in favor of the current project.

5. Major Risks

   Risks include any uncertain or unplanned events that may affect the viability and potential success of the project.

6. Project Timescales

   Timescales reflect the length or duration of a project and the time over which its benefits will be realized.

7. Project Costs

   Project costs are investment and other development costs for a project.

4.4.2 Business Justification and the Project Lifecycle

Business justification is first assessed prior to a project being initiated and is continuously verified throughout the project lifecycle. The following steps capture how business justification is determined:

1. Assess and Present a Business Case

   Business justification for a project is typically analyzed and confirmed by the Product Owner. It is documented and presented in the form of a project Business Case prior to Initiate phase and involves considering the various factors specified in section 4.4.1. Once documented, the Product Owner should create a Project Vision Statement and obtain approval of the Project Vision Statement from the key decision-makers in the organization. Generally, this consists of executives and/or some form of a project or program management board.

2. Continuous Value Justification

   Once the decision makers approve the Project Vision Statement, it is then baselined and forms the business justification. The business justification is validated throughout project execution, typically at predefined intervals or milestones, such as during portfolio, program, and Prioritized Product

Backlog Review Meetings and when major issues and risks that threaten project viability are identified. This could happen in several Scrum processes including *Conduct Daily Standup* and *Groom Prioritized Product Backlog*. Throughout the project, the Product Owner should keep the business justification in the Project Vision Statement updated with relevant project information to enable the key decision makers to continue making informed decisions.

3. Confirm Benefits Realization

   The Product Owner confirms the achievement of organizational benefits throughout the project, as well as upon completion of the User Stories in the Prioritized Product Backlog. Benefits from Scrum projects are realized during *Demonstrate and Validate Sprint*, *Retrospect Sprint*, *Ship Deliverables* and *Retrospect Project* processes.

Figure 4-3 summarizes the steps to determine business justification.

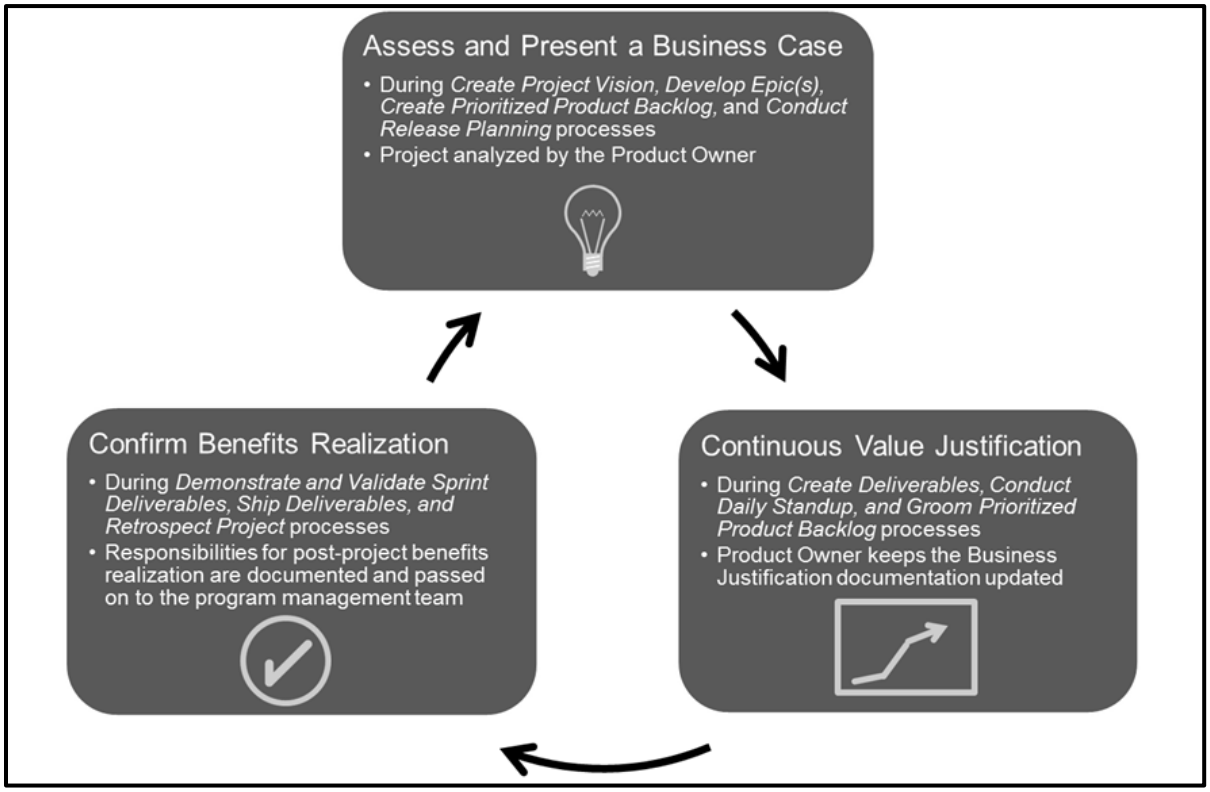


Figure 4-3: Business Justification and the Project Lifecycle

4.5 Business Justification Techniques

The following sections deal with some of the tools used to assess and evaluate business justification, as well as some other aspects associated with project justification and project selection. It is not necessary, or even recommended to use every available technique for every project. Some techniques are not appropriate depending on the specific project, and techniques may be used to assess projects individually or to compare the expected value of multiple projects.

The Scrum Guidance Body (SGB), which can be a panel of experts or a set of documents on organizational standards and procedures, defines the guidelines and metrics that will be used to assess business value. Each respective Product Owner, however, is responsible for performing the activities that verify and track business value for his or her respective projects, programs, or portfolios.

4.5.1 Estimation of Project Value

The value to be provided by business projects can be estimated using various methods such as Return on Investment (ROI), Net Present Value (NPV), and Internal Rate of Return (IRR).

1. Return on Investment (ROI)

   Return on Investment (ROI) when used for project justification, assesses the expected net income to be gained from a project. It is calculated by deducting the expected costs or investment of a project from its expected revenue and then dividing this (net profit) by the expected costs in order to get a return rate. Other factors such as inflation and interest rates on borrowed money may be factored into ROI calculations.

   ROI formula:

   ROI = (Project Revenue - Project Cost) / Project Cost

   Example: The ROI for a project that will cost $125,000 to develop, with expected financial benefits estimated at $300,000 is calculated as follows:

   ROI = ($300,000 - $125,000) / $125,000 = 1.4

   Therefore, the ROI is 1.4 times the investment (or 140%).

   Frequent product or service increments, is a key foundation of Scrum that allows earlier verification of ROI. This aids in assessing the justification of continuous value.

2. Net Present Value (NPV)

Net Present Value (NPV) is a method used to determine the current net value of a future financial benefit, given an assumed inflation or interest rate. In other words, NPV is the total expected income or revenue from a project, minus the total expected cost of the project, taking into account the time-value of money.

Example: Which of the following two projects is better to select if NPV is used as the selection criterion?

- Project A has a NPV of $1,500 and will be completed in 5 years.
- Project B has a NPV of $1,000 and will be completed in 1 year.

Solution: Project A, since its NPV is higher; the fact that Project B has a shorter duration than Project A is not considered here, because time is already accounted for in the NPV calculations (i.e., due to the fact that it is the current, not future value that is being considered in the calculation).

3. Internal Rate of Return (IRR)

Internal Rate of Return (IRR) is a discount rate on an investment in which the present value of cash inflows is made equal to the present value of cash outflows for assessing a project's rate of return. When comparing projects, one with a higher IRR is typically better.

Though IRR is not used to justify projects as often as some other techniques, such as NPV, it is an important concept to know.

Example: Based on IRR, which project is most desirable?

- Project A, which has an IRR of 15% and will be completed in 5 years.
- Project B, which has an IRR of 10% and will be completed in 1 year.

Solution: Project A, since its IRR is higher; the fact that Project B has a shorter duration than Project A is not considered here because time is already taken into account in the IRR calculations (i.e., as with NPV, it is the current, not future value that is used to determine the IRR).

4.5.2 Planning for Value

After justifying and confirming the value of a project, the Product Owner should consider the organizational policies, procedures, templates, and general standards dictated by the Scrum Guidance Body (or similar organizational project board or office) when planning a project; at the same time maximizing Value-driven Delivery. The onus for determining *how* the value is created falls on the stakeholders (sponsor, customers, and/or users), while the Scrum Team concentrates on *what* is to be developed. Some common tools recommended by a Scrum Guidance Body might include the following:

1. Value Stream Mapping

   Value Stream Mapping uses flowcharts, to illustrate the flow of information needed to complete a process. This technique may be used to streamline a process by helping to determine non-value-added elements.

2. Customer Value-based Prioritization

   Customer Value-based Prioritization places primary importance on the customer and strives to implement User Stories with the highest value first. Such high value User Stories are identified and moved to the top of the Prioritized Product Backlog.

   A team can use a variety of prioritization schemes to determine high-value features.

   a. Simple Schemes

      Simple schemes involve labeling items as Priority “1”, “2”, “3” or “High”, “Medium” and “Low” and so on. Although this is a simple and straightforward approach, it can become problematic because there is often a tendency to label everything as Priority “1” or “High”. Even “High,” “Medium,” and “Low” prioritization schemes can encounter similar difficulties.

   b. MoSCoW Prioritization

      The MoSCoW prioritization scheme derives its name from the first letters of the phrases “Must have,” “Should have,” “Could have,” and “Won’t have”. This prioritization method is generally more effective than simple schemes. The labels are in decreasing order of priority with “Must have” features being those without which the product will have no value and “Won’t have” features being those that, although they would be nice to have, are not necessary to be included.

   c. Monopoly Money

      This technique involves giving the customer “monopoly money” or “false money” equal to the amount of the project budget and asking them to distribute it among the User Stories under consideration. In this way, the customer prioritizes based on what they are willing to pay for each User Story.

d. 100-Point Method

The 100-Point Method was developed by Dean Leffingwell and Don Widrig (2003). It involves giving the customer 100 points they can use to vote for the features that they feel are most important.

e. Kano Analysis

The Kano analysis was developed by Noriaki Kano (1984) and involves classifying features or requirements into four categories based on customer preferences:

1. *Exciters/Delighters:* Features that are new, or of high value to the customer
2. *Satisfiers:* Features that offer value to the customer
3. *Dissatisfiers:* Features which, if not present, are likely to cause a customer to dislike the product, but do not affect the level of satisfaction if they are present
4. *Indifferent:* Features that will not affect the customer in any way and should be eliminated

Figure 4-4 depicts an illustration of Kano Analysis.

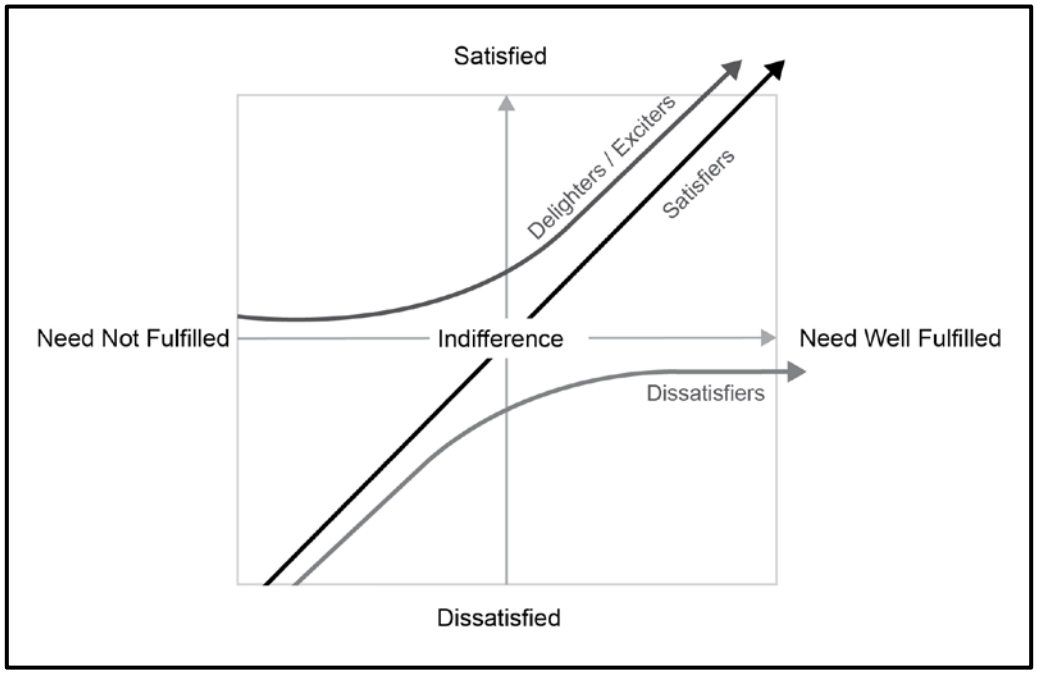


Figure 4-4: Kano Analysis

Interestingly, features usually move down the classification list over time; customers will come to expect features (e.g., cameras on phones) and these features will move from being exciters/delighters to satisfiers and eventually to dissatisfiers.

4.5.3 Relative Prioritization Ranking

A simple listing of User Stories in order of priority is an effective method for determining the desired User Stories for each iteration or release of the product or service. The purpose is to create a simple, single list with the goal of prioritizing features, rather than being distracted by multiple prioritization schemes.

This simple list also provides a basis for incorporating changes and identified risks when necessary. Each change or identified risk can be inserted in the list based on its priority relative to the other User Stories in the list. Typically, new changes will be included at the expense of features that have been assigned a lower priority.

Defining the Minimum Marketable Features (MMF) is extremely important during this process, so that the first release or iteration happens as early as possible, leading to increased ROI. Normally, these User Stories would rank highest in priority.

4.5.4 Story Mapping

This is a technique to provide a visual outline of the product and its key components. Story Mapping, first formulated by Jeff Patton (2005), is commonly used to illustrate product roadmaps.

Story maps depict the sequence of product development iterations and map out which features will be included in the first, second, third, and subsequent releases.

4.6 Continuous Value Justification

Business value should be assessed regularly to determine whether the justification or viability of executing the project continues to exist. Frequent assessment of investment in the project relative to business value being created qualifies the continued viability of a project. The expected requirements from the project may change frequently, which can impact both project investment and value creation. A key aspect of Scrum is its ability to quickly adjust to chaos created by a rapidly changing business model. In projects with ambiguous user requirements and significant potential for frequent changes, Scrum provides considerable advantages over other development models.

Monitoring the rate of delivering value is an important requirement for Scrum projects. Periodically tracking and reporting the creation of value assists in assessing project status and provides important information to the customer and other stakeholders.

4.6.1 Earned Value Analysis

Although commonly used, tools such as bar charts and Gantt Charts have limitations in tracking and reporting progress when it comes to project performance. Earned Value Analysis (EVA) is used for this purpose.

EVA analyzes actual project performance against planned performance at a given point in time. For tracking techniques to be effective, the initial baseline project plan needs to be accurate. EVA often uses graphs and other visuals (e.g., S-curve), as a way to depict project status information.

Earned Value Analysis measures current variances in the project's schedule and cost performance and forecasts the final cost based on the determined current performance. EVA is typically done at the end of each Sprint after the User Stories in Sprint Backlog are completed.

Table 4-1 summarizes the formulas used in Earned Value Analysis.

Term Definition	Acronym	Formula
Planned Value	PV	
Earned Value	EV	
Actual Cost	AC	
Budget at Completion	BAC	
Schedule Variance	SV	EV - PV
Cost Variance	CV	EV - AC
Schedule Performance Index	SPI	EV / PV
Cost Performance Index	CPI	EV / AC
Percent Complete	% Complete	(EV / BAC) x 100
Estimate at Completion 1. Estimating assumptions not valid 2. Current Variances are atypical 3. Current Variances are typical	EAC	1. AC + ETC 2. AC + BAC - EV 3. BAC / CPI
Estimate to Complete	ETC	EAC - AC
Variance at Completion	VAC	BAC - EAC

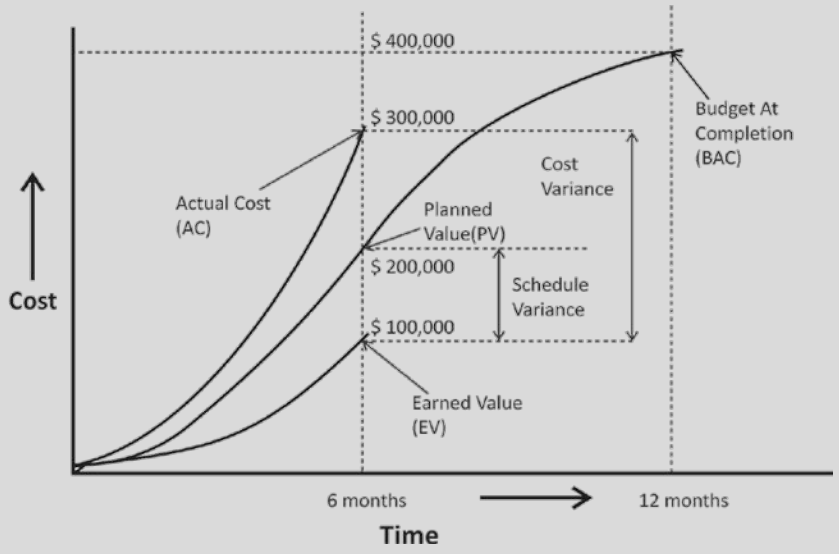
Table 4-1: Earned Value Formulas

Example: A website with 4,000 web pages needs to be developed—we assume that every web page takes the same time to complete, and that each web page is a unique User Story of equal priority in the Prioritized Product Backlog. The estimated cost of completing the project is $400,000 and the time limit for the project is 12 months. After 6 months, $300,000 has been spent and the work completed is 1,000 web pages.

What have we been provided with?

- Budget at Completion (BAC) = $400,000 (Cost Baseline for the project)
- Planned Value (PV) = $200,000 (since we planned to complete 2,000 web pages)
- Earned Value (EV) = $100,000 (value of 1,000 web pages that are complete)
- Actual Cost (AC) = $300,000 (what has been spent so far)

S-curve for the data:



Formulas:

- Schedule Variance (SV) = EV - PV = $100,000 - $200,000 = - $100,000
- Cost Variance (CV) = EV - AC = $100,000 - $300,000 = - $200,000
    - The negative variances in our project indicate that we are behind schedule and over budget.
- Schedule Performance Index (SPI) = EV / PV = $100,000 / $200,000 = 0.5
    - SPI < 1 indicates that the work completed so far is only 50% of what we planned to have completed at 6 months.
- Cost Performance Index (CPI) = EV / AC = $100,000 / $300,000 = 0.33
    - CPI < 1 indicates that we are only getting 33% of work done for the amount of money being spent.
- Percent Complete = EV / BAC x 100 = $100,000 / $400,000 x 100 = 25%
    - So, 25% of the work on the project is complete at this point in time.

4.6.2 Cumulative Flow Diagram (CFD)

A Cumulative Flow Diagram (CFD) is a useful tool for reporting and tracking project performance. It provides a simple, visual representation of project progress at a particular point in time. It is usually used to provide a higher level status of the overall project and not daily updates for individual Sprints.

Figure 4-5 is an example of a CFD for a large project. It shows how many User Stories are yet to be created, in process of being created, and have been created. As customer requirements change, there is a change in the Cumulative User Stories which have to be delivered. Change points 1 and 2 are where the Product Owner removed existing user Stories in the Risk Adjusted Prioritized Product Backlog and Change points 3 and 4 are where the Product Owner added new User Stories in the Risk Adjusted Prioritized Product Backlog

This type of diagram can be a great tool for identifying roadblocks and bottlenecks within processes. For example, if the diagram shows one band becoming narrower while the previous band is becoming wider over time, there may be a bottleneck and changes may be needed to increase efficiency and/or improve project performance.

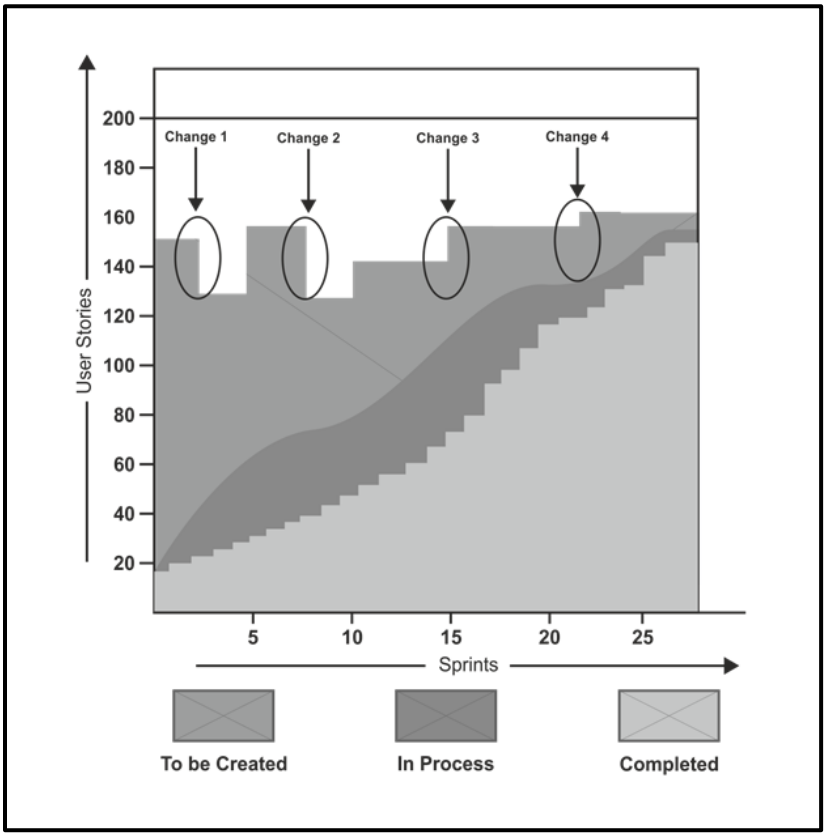


Figure 4-5: Sample Cumulative Flow Diagram (CFD)

4.7 Confirm Benefits Realization

Throughout a project, it is important to verify whether benefits are being realized. Whether the products of a Scrum project are tangible or intangible, appropriate verification techniques are required to confirm that the team is creating the deliverables that will achieve the benefits and value defined at the beginning of the project.

4.7.1 Prototypes, Simulations, and Demonstrations

Demonstrating prototypes to customers and simulating their functionalities are commonly used techniques for confirming value.

Often, after using the features or having them demonstrated, customers can more clearly determine whether the features are adequate and suitable for their needs. They might realize a need for additional features, or may decide to modify previously defined feature requirements. In product development, this customer experience has come to be known as IKIWISI (I'll Know It When I See It).

Through demonstrations or access to early iterations, customers can also evaluate to what degree the team has successfully interpreted their requirements and met their expectations.

4.8 Summary of Responsibilities

Role	Responsibilities
Scrum Team	• Ensures that project deliverables are completed in accordance with agreed Acceptance Criteria • Performs Continuous Value Justification for projects
Product Owner/ Chief Product Owner	• Ensures value delivery for projects • Maintains the business justification for projects • Confirms and communicates project benefits to stakeholders
Scrum Master/ Chief Scrum Master	• Ensures the desired outcomes of the project are communicated to and understood by the Scrum Team • Performs Continuous Value Justification for projects
Program Product Owner	• Ensures value delivery for programs • Creates the business justification for programs • Provides value guidance for projects within a program • Approves the business justification of projects within a program
Program Scrum Master	• Ensures the desired outcomes of the program are communicated and understood • Performs Continuous Value Justification for programs
Portfolio Product Owner	• Ensures value delivery for portfolios • Creates the business justification for portfolios • Provides value guidance for programs within portfolios • Approves the business justification of programs within a portfolio
Portfolio Scrum Master	• Ensures the desired outcomes of the portfolio are achieved • Performs Continuous Value Justification for portfolios
Stakeholder(s)	• Helps prioritize User Stories and requirements in the Prioritized Product Backlog • Communicates with Scrum Team and confirms realization of value at the end of every Sprint, Release, and the project
Scrum Guidance Body	• Establishes overall guidelines and metrics for evaluating value • Acts in a consulting capacity and provides guidance for projects, programs, and portfolios as required

Table 4-2: Summary of Responsibilities Relevant to Business Justification

4.9 Scrum vs. Traditional Project Management

Traditional projects emphasize on extensive upfront planning and adherence to the project plan created by the project manager. Usually, changes are managed through a formal change management system and value is created at the end of the project when the final product is delivered.

In Scrum projects, extensive long-term planning is not done prior to project execution. Planning is done in an iterative manner before each Sprint. This allows quick and effective response to change, which results in lower costs and ultimately increased profitability and Return on Investment (ROI). Moreover, Value-Driven Delivery (section 4.3) is a key benefit of the Scrum framework and provides significantly better prioritization and quicker realization of business value. Because of the iterative nature of Scrum development, there is always at least one release of the product with Minimum Marketable Features (MMF) available. Even if a project is terminated, there are usually some benefits or value created prior to termination.

APPENDIX B. *SBOK™ GUIDE* AUTHORS AND CONTRIBUTORS

This appendix lists the names of those individuals who contributed to the development and production of the *SBOK™ Guide*.

SCRUMstudy™ is grateful to all these individuals for their continuous support and acknowledges their contributions towards the development of the *SBOK™ Guide*.

B.1 Lead Author

Tridibesh Satpathy

B.2 Coauthors and Subject Matter Expert Committee

Winfried Hackmann
Gaynell Malone
J. Drew Nations
Buddy Peacock
Deepak R
Ruth Kim

B.3 Contributors and Reviewers

James Pruitt
Elizabeth Lynne Warren
Syed Ashraf
Melvin Wofford Jr.
Srinivas Reddy Kandi
Girish Kulkarni
Abdelnaser Dwaikat
Anu Ravi
Morris Feigel
Ian Glenister
Jo Pereira

Saurabh Gupta
Nikhil Bhargava
Simon Robertson
Dusan Kamenov
Prof. Dr. Akram Hassan
Meena Elangovan
Olumide Idowu
Vinod Kumar
Michael Rauch
Joe Schofield

B.4 Prior Editions

Coauthors and Subject Matter Experts

R-A Alves
Winfried Hackmann
Quincy D. Jordan
Gaynell Malone
J. Drew Nations
Buddy Peacock
Karen Lyncook
Jaimie M. Rush
Elizabeth Lynne Warren
Ruth Kim
Mehul Doshi
Gaurav Garg
Ajey Grandhem
Sayan Guha
Vinay Jagannath
Deepak Ramaswamy
Ahmed Touseefullah Siddiqui

Reviewers and Edit Team

Corey T. Bailey
Sohini Banerjee
Vince Belanger
Bobbie Green
Magaline D. Harvey
Ravneet Kaur
Robert Lamb
Mimi LaRaque
Melissa Lauro
Richard Mather
Lachlan McGurk
Madhuresh Kumar Mishra
Neha Mishra
Yogaraj Mudalgi
Jose Nunez
Obi Nwaojigba
Bryan Lee Perez
James Pruitt
Charles J. Quansah
Frank Quinteros
Nadra Rafee
Tommie L. Sherrill
Barbara Siefken
Sandra A. Strech
Frances Mary Jo Tessler
Chrys Thorsen
Mike Tomaszewski
Ron Villmow

APPENDIX C. THIRD EDITION UPDATES

This appendix provides a summary of updates implemented in the *SBOK™ Guide*—Third Edition as compared to the previous edition.

C.1 Summary of Changes

The scope of updates made for the *SBOK™ Guide*—Third Edition primarily focused on the following major areas:

- Improved and expanded description of roles and responsibilities in the Scrum Framework, particularly as they relate to large projects, programs, and portfolios.
- Clarification and streamlining of the processes identified for the Plan and Estimate phase. This included simplification of the meetings involved in these processes.
- Additional content covering how to scale Scrum for large projects and at the enterprise level.

General improvements were also made throughout the text to ensure information was accurate, clear, and complete. This included updates to tables and figures as appropriate.

C.2 Third Edition Updates by Chapter

Chapter	Key Changes Made
1	• Improved consistency and clarity. • Added reference to two new certifications, SSMC™ and SSPOC™ (section 1.3). • Updated Scrum Processes (section 1.4.4) to reflect new process names for Plan and Estimate phase (see Chapter 9). Also, added the processes discussed in Chapters 13 and 14 for Scaling Scrum for Large Projects and Scaling Scrum for the Enterprise.
2	• Simplified the verbiage for the Three Daily Questions in the "Conduct Daily Standup" process to be more generic to meeting time of day (section 2.7.1) • Provided more detailed description of Sprint Planning Meeting (section 2.7.1)

Chapter	Key Changes Made
3	• In general, this chapter was restructured to consolidate the descriptions of roles and responsibilities under the core Scrum roles: Product Owner (section 3.4), Scrum Master (section 3.5) and Scrum Team (section 3.6). This includes expanded definitions, particularly for roles related to large projects, programs and portfolios. • Summary of Responsibilities (section 3.8) updated to include Chief Product Owner and Chief Scrum Master roles.
4	• Summary of Responsibilities (section 4.8) updated to include Chief Product Owner and Chief Scrum Master roles.
5	• Improved description of Definition of Done (section 5.4.2) and Minimum Done Criteria (section 5.4.3) • Summary of Responsibilities (section 5.6) updated to include Chief Product Owner and Chief Scrum Master roles.
6	• Summary of Responsibilities (section 6.7) updated to include Chief Product Owner and Chief Scrum Master roles.
7	• Summary of Responsibilities (section 7.7) updated to include Chief Product Owner and Chief Scrum Master roles.
8	• Moved descriptions for Program Product Owner and Program Scrum Master to Chapter 3 for consistency. • Minor changes to update terminology and figures to match updates made in other chapters.
9	• The "Approve, Estimate and Commit User Stories" process was replaced by the following two processes: "Estimate User Stories" (section 9.2) and "Commit User Stories" (section 9.3). This was done to provide improved clarity to the inputs, tools and outputs relevant to the activities performed in these processes. • A new tool, "Estimation Methods" was defined to consolidate many of the estimation techniques called out individually in the previous edition (section 9.2.2.3, 9.5.2.3). • The "Create Tasks" process was renamed to "Identify Tasks" (section 9.4), to clarify that tasks are defined or identified based on the previously Committed User Stories. • Inputs, tools and outputs for all the processes in the Plan and Estimate phase were evaluated and adjusted for correctness.

Chapter	Key Changes Made
10	• The verbiage for the Three Daily Questions in the "Conduct Daily Standup" process was updated to be more generic to meeting time of day (section 10.2.2.2). • Minor changes to update terminology and figures to match updates made in other chapters.
11	• Removed "Convene Scrum of Scrums" process. This is now addressed in Chapter 13, Scaling Scrum for Large Projects. • Minor changes to update terminology and figures to match updates made in other chapters.
12	• Minor changes to update terminology and figures to match updates made in other chapters.
13	• Scaling Scrum for Large Projects — entire chapter added as new content.
14	• Scaling Scrum for the Enterprise — entire chapter added as new content.

REFERENCES

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., DeCarlo, D., Fitzgerald, D., Highsmith, J., Jepsen, O., Lindstrom, L., Little, T., McDonald, K., Pixton, P., Smith, P., and Wysocki, R. (2005) "Declaration of Interdependence," accessed September 2013, http://www.pmdoi.org/.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001) "Manifesto for Agile Software Development," accessed September 2013, http://agilemanifesto.org/.

Fellers, G. (1994) *Why Things Go Wrong: Deming Philosophy In A Dozen Ten-Minute Sessions*. Gretna, LA: Pelican Publishing.

Greenleaf, R. K. (1977) *Servant Leadership: A Journey into the Nature of Legitimate Power and Greatness. Mahwah*, NJ: Paulist Press.

Kano, N., Seraku, N., Takahashi, F., and Tsuji, S. (1984) "Attractive Quality and Must Be Quality." *Quality*, 14 (2): 39–48.

Leffingwell, D. and Widrig, D. (2003) *Managing Software Requirements: A Use Case Approach, 2nd ed*. Boston: Addison-Wesley.

Maslow, A. H. (1943) "A Theory of Human Motivation." *Psychological Review*, 50 (4): 370–396.

McGregor, D. (1960) *The Human Side of Enterprise*. New York: McGraw-Hill.

Patton, J. (2005) "It's All in How You Slice." Better *Software*, January: 16–40.

Spears, L. C. (2010) "Character and Servant Leadership: Ten Characteristics of Effective, Caring Leaders." *The Journal of Virtues & Leadership*, 1 (1): 25–30.

Takeuchi, H. and Nonaka, I. (1986) "The New New Product Development Game." *Harvard Business Review*, January–February: 137–146.

GLOSSARY

100-Point Method

The 100-Point Method was developed by Dean Leffingwell and Don Widrig (2003). It involves giving the customer 100 points they can use to vote for the features that they feel are most important.

Accepted Deliverables

Deliverables which meet the User Story Acceptance Criteria are accepted by the Product Owner. These are considered Accepted Deliverables that may be released to the customer if they so desire.

Actionable Escalations

The Scrum Guidance Body may determine that some company policies do not allow teams to get the maximum benefits from the application of Scrum. In such a case, an escalation needs to be triggered in order to get approval for a policy change.

Adaptation

Adaptation happens as the Scrum Core Team and Stakeholder(s) learn through transparency and inspection and then adapt by making improvements in the work they do.

Affinity Estimation

Affinity Estimation is a technique used to quickly estimate a large number of User Stories using categories. The categories can be small, medium, or large, or they may be numbered using story point values to indicate relative size. Some key benefits of this approach are that the process is very transparent, visible to everyone, and is easy to conduct.

Agreed Actionable Improvements

Agreed Actionable Improvements are the primary output of the *Retrospect Sprint* process. They are the list of actionable items that the team has come up with to address problems and improve processes in order to enhance their performance in future Sprints.

Approved Change Requests

Approved Change Requests are changes that have been approved to be included in the Prioritized Product Backlog. At times, Approved Change Requests may originate from the program or portfolio managers and would be inputs to be added to the list of approved project changes for implementation in future Sprints.

Assertive Leader

Assertive leaders confront issues and display confidence to establish authority with respect.

Assessments/Benchmarking Results

Assessment/benchmarking help set a minimum standard when creating a product or service and lead to changed Done Criteria. Sometimes they may also provide impetus for a Program or Portfolio Product Owner to develop new User Stories to implement the best practices.

Assigned Action Items and Due Dates

Once the Agreed Actionable Improvements have been elaborated and refined, action items to implement the improvements may be considered by the Scrum Team. Each action item will have a defined due date for completion.

Autocratic Leader

Autocratic leaders make decisions on their own, allowing team members little, if any, involvement or discussion before a decision is made. This leadership style should only be used on rare occasions.

Automated Software Tools

Automated Software Tools are software tools used for scheduling, information collection, and distribution.

Better Team Coordination

The Scrum of Scrums Meeting facilitates coordination of work across multiple Scrum Teams. This is especially important when there are tasks involving inter-team dependencies. Incompatibilities and discrepancies between the work and deliverables of different teams are quickly exposed. This forum also gives teams the opportunity to showcase their achievements and give feedback to other teams.

Benchmarking

An enterprise should benchmark its own practices against other companies regularly in order to keep up with the competition. Benchmarking is the process of comparing an organization's business processes and performance metrics to those of leading companies in the same or other industries.

Brainstorming

Sessions where relevant stakeholders and members of the Scrum Core Team openly share ideas through discussions and knowledge sharing sessions, which are normally conducted by a facilitator.

Business Justification

Business Justification demonstrates the reasons for undertaking a project. It answers the question "Why is this project needed?" Business justification drives all decision making related to a project.

Business Needs

Business needs are those business outcomes that the project is expected to fulfill, as documented in the Project Vision Statement.

Business Requirements

Business Requirements define what must be delivered to fulfill business needs and provide value to stakeholders. The sum of all the insights gained through various tools such as user or customer interviews, questionnaires, JAD sessions, Gap Analysis, SWOT Analysis, and other meetings, helps get a better perspective about the business requirements and helps in creating the Prioritized Product Backlog.

Change Request(s)

Request for changes are usually submitted as Change Requests. Change Requests remain in an unapproved status until they are formally approved.

Chief Product Owner

In the case of large projects, the Chief Product Owner prepares and maintains the overall Prioritized Product Backlog for the project. He or she coordinates work among the Product Owners of the Scrum Teams. The Product Owners, in turn, manage their respective parts of the Prioritized Product Backlog.

Chief Scrum Master

In case of large projects, the Chief Scrum Master is responsible for moderating the Scrum of Scrums (SoS) Meeting and removing impediments that affect multiple teams.

Coaching/Supportive Leader

Coaching and supportive leaders issue instructions and then support and monitor team members through listening, assisting, encouraging, and presenting a positive outlook during times of uncertainty.

Collaboration

Collaboration in Scrum refers to the Scrum Core Team working together and interfacing with the stakeholders to create and validate the deliverables of the project to meet the goals outlined in the Project Vision. Collaboration occurs when a team works together to play off each other's contributions to produce something greater.

Collaboration Plan

Collaboration is an extremely important element in Scrum and the Collaboration Plan outlines how the various decision makers, stakeholders, and team members engage and collaborate with each other.

Colocation

Colocation is having all Scrum Core Team members located in the same work place leveraging the advantages of better coordination, problem-solving, knowledge sharing, and learning.

Commit User Stories

In this process, the Scrum Team commits to deliver User Stories approved by the Product Owner for a Sprint. The result of this process would be Committed User Stories.

Communication Plan

This plan specifies the records that must be created and maintained throughout the project. A variety of methods are used to convey important project information to stakeholders. The Communication Plan defines these methods as well as who is responsible for the various communication activities.

Company Human Resource Plans

Company Human Resource Plans, broadly, provide information on when particular personnel will be available for various projects, programs, and portfolios. They also provide information on plans for hiring personnel required for future efforts.

Company Mission

The Company Mission provides a framework for formulating the strategies of a company or organization that guides their overall decision making.

Company Policies

Company policies are a set of principles, rules, and guidelines formulated or adopted by an organization. Changing company policies would affect created User Stories as they have been created following existing policies.

Company Vision

Understanding the Company Vision helps the project keep its focus on the organization's objectives and the future potential of the company. The Product Owner can take guidance and direction from the Company Vision to create the Project Vision Statement.

Conduct Daily Standup

Conduct Daily Standup is a process in which a highly focused, Time-boxed meeting is conducted every day. This meeting is referred to as a Daily Standup Meeting, which is a forum for the Scrum Team to update each other on their progress and any impediments they may be facing.

Conduct Release Planning

In this process, the Scrum Core Team reviews the high-level 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 Stakeholder(s). The Length of Sprints is also determined in this process.

Conflict Management

Conflict Management techniques are used by team members to manage any conflicts that arise during a Scrum project. Sources of conflict often include schedules, priorities, resources, reporting hierarchy, technical issues, procedures, personality, and costs.

Continuous Improvement

Continuous Improvement is a Scrum approach in which the team learns from experience and stakeholder engagement to constantly keep the Prioritized Product Backlog updated with any changes in requirements.

Continuous Value Justification

Continuous Value Justification refers to assessment of business value regularly to determine whether the justification or viability of executing the project continues to exist.

Core Role(s)

Core Roles are those roles which are mandatorily required for producing the product of the project, are committed to the project, and ultimately are responsible for the success of each Sprint within the project and of the project as a whole.

Create Deliverables

Create Deliverables is the process in which the Scrum Team works on the tasks in the Sprint Backlog to create Sprint Deliverables.

Create Prioritized Product Backlog

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

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.

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.

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.

Cumulative Flow Diagram (CFD)

A Cumulative Flow Diagram (CFD) is a useful tool for reporting and tracking project performance. It provides a simple, visual representation of project progress at a particular point in time. It is usually used to provide a higher level status of the overall project and not daily updates for individual Sprints.

Customer

The Customer is an individual or the organization that acquires the project's product, service, or other result. For any organization, depending on the project, there can be both internal customers (i.e., within the same organization) or external customers (i.e., outside of the organization).

Customer Value-based Prioritization

Customer Value-based Prioritization places primary importance on the customer and strives to implement User Stories with the highest value first. Such high value User Stories are identified and moved to the top of the Prioritized Product Backlog.

Daily Standup Meeting

The Daily Standup Meeting is a short daily meeting, Time-boxed to 15 minutes. The team members gather to report their progress by answering the following three questions:

1. What have I done since the last meeting?
2. What do I plan to do before the next meeting?
3. What impediments or obstacles (if any) am I currently facing?

Decomposition

Decomposition is a tool whereby high-level tasks are broken down into lower level, more detailed tasks. The User Stories are decomposed into tasks by members of the Scrum Team. Prioritized Product Backlog User Stories should be sufficiently decomposed to a level that provides the Scrum Team adequate information to create deliverables from the Tasks mentioned in the Task List.

Delegating Leader

Delegating Leaders are involved in the majority of decision making; however, they delegate some planning and decision-making responsibilities to team members, particularly if they are competent to handle tasks. This leadership style is appropriate in situations where the leader is in tune with specific project details and when time is limited.

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.

Dependency Determination

Once the Scrum Team has selected User Stories for a given Sprint, they should then consider any dependencies, including those related to the availability of people as well as any technical dependencies. Properly documenting dependencies helps the Scrum Teams determine the relative order in which tasks should be executed to create the Sprint Deliverables. Dependencies also highlight the relationship and interaction between tasks both within the Scrum Team working on a given Sprint and with other Scrum Teams in the project.

Design Patterns

Design Patterns provide a formal way of recording a resolution to a design problem in a specific field of expertise. These patterns record both the process used and the actual resolution, which can later be reused to improve decision making and productivity.

Develop Epic(s)

In this process, the Project Vision Statement serves as the basis for developing large, high-level, unrefined User Stories referred to as Epics. User Group Meetings may be held to *Develop Epic(s).*

Development in Phases Contract

This contract makes funding available each month or each quarter after a release is successfully completed. It gives incentive to both customer and supplier and ensures that the monetary risk of the customer is limited to that particular time period since unsuccessful releases are not funded.

Directing Leader

Directing Leaders instruct team members regarding what tasks are required and when and how they should be performed.

Discretionary Dependencies

Discretionary Dependencies are dependencies that are placed into the workflow by choice. Typically, discretionary dependencies are determined by the Scrum Team based on past experiences or best practices in a particular field or domain.

Done Criteria

Done Criteria are a set of rules that are applicable to all User Stories. A clear definition of Done is critical, because it removes ambiguity from requirements and helps the team adhere to mandatory quality norms. This clear definition is used to create the Done criteria that are an output of the *Create Prioritized Product Backlog* process. A User Story is considered done when it is demonstrated to and approved by the Product Owner who judges it on the basis of the Done Criteria and the User Story Acceptance Criteria.

Earned Value Analysis

Earned Value Analysis analyzes actual project performance against planned performance at a given point in time. It measures current variances in the project's schedule and cost performance and forecasts the final cost based on the determined current performance.

Effort Estimated Task List

The Effort Estimated Task List is a list of tasks associated with the committed User Stories included in a Sprint. Estimated effort is expressed in terms of the estimation criteria agreed upon by the team. The Effort Estimated Task List is used by the Scrum Team during Sprint Planning Meetings to create the Sprint Backlog and the Sprint Burndown Chart.

Empirical Process Control

An Empirical Process Control model helps make decisions based on observation and experimentation rather than on detailed upfront planning. It relies on the three main ideas of transparency, inspection, and adaptation.

Environment(s)

This refers to the identification and documentation of all the environments required to develop and test the project's deliverables.

Environment Meeting

This meeting is conducted to identify the types and number of environments required to develop, manage, and test the project's deliverables. In this meeting, the resources required to establish the required environments are also discussed.

Environment Planning Meeting

An Environment Plan Meeting is used to define a schedule/calendar of how the Scrum Teams will share environments.

Environment Schedule

Environment Schedule is a schedule/calendar of how the Scrum Teams will share the environments. It provides allocated days and time periods for each team to use each environment.

Epic(s)

Epic(s) are written in the initial stages of the project when most User Stories are high-level functionalities or product descriptions and requirements are broadly defined. They are large, unrefined User Stories in the Prioritized Product Backlog.

Estimate Range

Estimates for projects should be presented in ranges. Precise figures may give an impression of being highly accurate when in fact they may not be. In fact, estimates by definition are understood not to be precisely accurate. Estimate ranges should be based on the level of confidence the team has in each estimate.

Estimate Tasks process

In this process, the Scrum Core Team, in a Task Estimation Workshop, estimates the effort required to accomplish each task in the Task List. The output of this process is an Effort Estimated Task List.

Estimate User Stories

In this process, the Product Owner clarifies User Stories in order for the Scrum Master and Scrum Team to estimate the effort required to develop the functionality described in each User Story.

Estimation Criteria

The primary objective of using Estimation Criteria is to maintain relative estimation sizes and minimize the need for re-estimation. Estimation Criteria can be expressed in numerous ways, with two common examples being story points and ideal time.

Expected Monetary Value

This is a risk assessment technique where the potential financial impact of a risk is determined based on its Expected Monetary Value (EMV). EMV is calculated by multiplying the monetary impact by the risk's probability, as approximated by the customer.

Explorer—Shopper—Vacationer—Prisoner (ESVP)

This is an exercise that can be conducted at the start of the Retrospect Sprint Meeting to understand the mind-set of the participants and set the tone for the meeting. Attendees are asked to anonymously indicate which best represents their outlook in the meeting.

External Dependencies

External dependencies are those related to tasks, activities, or products that are outside the scope of the work to be executed by the Scrum Team, but are needed to complete a project task or create a project deliverable. External dependencies are usually outside the Scrum Team's control.

Fist of Five

Fist of Five is a simple and fast mechanism that can be used as an estimation practice, as well as a general group consensus building technique. After initial discussion on a given item for estimation, the Scrum Team members are each asked to vote on a scale of 1 to 5 using their fingers.

Focus Group Meetings

Focus groups assemble individuals in a guided session to provide their opinions, perceptions, or ratings of a product, service, or desired result. Focus group members have the freedom to ask questions to each other and to get clarifications on particular subjects or concepts. Through questioning, constructive criticism, and feedback, focus groups lead to a better quality product and thereby contribute to meeting the expectations of the users.

Form Scrum Team

The Scrum Team members are identified during this process. Normally the Product Owner has the primary responsibility of selecting team members, but he or she often does so in collaboration with the Scrum Master.

Forming Stage

Forming Stage is the first stage of team formation, often considered a fun stage because everything is new and the team has not yet encountered any difficulties with the project.

Four Questions per Team

A set of questions asked in each Scrum of Scrums (SoS) Meeting. Each Scrum Team representative will provide updates from his or her team which are usually provided in the form of answers to four specific questions.

1. What has my team been working on since the last meeting?
2. What will my team do until the next meeting?
3. What were other teams counting on our team to finish that remains undone?
4. What is our team planning on doing that might affect other teams?

Gap Analysis

Gap Analysis is a technique used to compare the current, actual state with some desired state and to determine how to bridge the gap between them.

Groom Prioritized Product Backlog

Groom Prioritized Product Backlog is a process in which the Prioritized Product Backlog is continuously updated and maintained.

Identify Environment

Identifying the number and types of environments needed because numerous Scrum Teams that will be starting and ending their Sprints on the same day.

Identify Scrum Master and Stakeholder(s) process

In this process, the Scrum Master and the stakeholders are identified using specific Selection Criteria.

Identify Tasks

In this process, the Committed User Stories are broken down into specific tasks and compiled into a Task List. This is done as part of the Sprint Planning Meeting.

Industry Standards

New industry standards or changes to existing standards need to be implemented in order to maintain a viable product or service. Therefore, related User Stories need to be included in the Prioritized Program and/or Portfolio Backlog and prioritized accordingly.

Impediment

An impediment is any hindrance or hurdle that reduces the productivity of the Scrum Team.

Implement Phase

The Implement Phase includes processes related to the execution of the tasks and activities to create a project's product.

Incentive and Penalty Contract

This contract is based on the agreement that the supplier will be rewarded with a financial incentive, if the project's products are delivered on time, but will incur financial penalties, if the delivery is late.

Incremental Delivery Contract

This contract includes inspection points at regular intervals. It helps the customer or stakeholders make decisions regarding product development periodically throughout the project at each inspection point. The customer can either accept the development of the product, decide to stop the development of the product, or request product modifications.

Index Cards

Index cards, often described as Story Cards, are used to track the User Stories throughout the project. This increases visibility and transparency and facilitates early discovery of any problems that may arise.

Initiate phase

This phase is composed of the processes related to initiation of a project: *Create Project Vision, Identify Scrum Master and Stakeholder(s), Form Scrum Team, Develop Epic(s), Create Prioritized Product Backlog*, and *Conduct Release Planning*.

Inspection

Inspection refers to the monitoring required to follow empirical process control, to ensure that the project deliverables conforms to the requirements.

Internal Dependencies

Internal dependencies are those dependencies between tasks, products, or activities that are under the control of the Scrum Team and within the scope of the work to be executed by the Scrum Team.

Internal Rate of Return (IRR)

Internal Rate of Return (IRR) is a discount rate on an investment in which the present value of cash inflows is made equal to the present value of cash outflows for assessing a project's rate of return. When comparing projects, one with a higher IRR is typically better.

Issues

Issues are generally well-defined certainties that are currently happening on the project, so there is no need for conducting a probability assessment as would be done for a risk.

Iterative Delivery

Iterative delivery is the phased delivery of value to the customer.

JAD Sessions

A Joint Application Design (JAD) session is a requirements gathering technique. It is a highly structured facilitated workshop which hastens the *Create Project Vision* process as it enables the Stakeholder(s) and other decision makers to come to a consensus on the scope, objectives, and other specifications of the project.

Joint Venture Contract

This contract is generally used when two or more parties partner to accomplish the work of a project. The parties involved in the project will both achieve some Return on Investment because the revenues or benefits generated will be shared between the parties.

Kano Analysis

Kano Analysis was developed by Noriaki Kano (1984) and involves classifying features or requirements into four categories based on customer preferences:

1. Exciters/Delighters
2. Satisfiers
3. Dissatisfiers
4. Indifferent

Laissez Faire Leader

A leadership style, where the team is left largely unsupervised and the leader does not interfere with their daily work activities. This often leads to a state of anarchy.

Large Core Team

The Large Core Team comprises the Chief Product Owner, Chief Scrum Master, Scrum Masters, Product Owners, and selected members of the Scrum Teams working on the large project.

Length of Sprint

Based on the various inputs including business requirements and the Release Planning Schedule, the Product Owner and the Scrum Team decide on the length of the Sprints for the project. Once determined, the length of the Sprint is usually fixed for the project. Length of Sprint is the time duration of the Sprints determined for a project.

Mandatory Dependencies

These dependencies are either inherent in the nature of the work, like a physical limitation, or may be due to contractual obligations or legal requirements.

Market Study

Market Study refers to the organized research, gathering, collation, and analysis of data related to customers' preferences for products. It often includes extensive data on market trends, market segmentation, and marketing processes.

Members Selection Criteria

Members Selection Criteria are created by stakeholders to define the Scrum Guidance Body members, their roles and responsibilities, the number of members, and their required skills and expertise.

Minimum Acceptance Criteria

Minimum Acceptance Criteria are declared by the business unit. They then become part of the Acceptance Criteria for any User Story for that business unit. Any functionality defined by the business unit must satisfy these Minimum Acceptance Criteria, if it is to be accepted by the respective Product Owner.

Mitigated Risks

Mitigated Risks refer to the risks that are successfully addressed or mitigated by the Scrum Team during the project.

Monopoly Money

Monopoly Money is a technique that involves giving the customer "monopoly money" or "false money" equal to the amount of the project budget and asking them to distribute it among the User Stories under consideration. In this way, the customer prioritizes based on what they are willing to pay for each User Story.

MoSCoW Prioritization

The MoSCoW Prioritization scheme derives its name from the first letters of the phrases "Must have," "Should have," "Could have," and "Won't have". The labels are in decreasing order of priority with "Must have" features being those without which the product will have no value and "Won't have" features being those that, although they would be nice to have, are not necessary to be included.

Net Present Value (NPV)

Net Present Value (NPV) is a method used to determine the current net value of a future financial benefit, given an assumed inflation or interest rate.

Non-core role

Non-core roles are those roles which are not mandatorily required for the Scrum project. They may include team members who are interested in the project, who have no formal role on the project team, may interface with the team, but may not be responsible for the success of the project.

Norming stage

The third stage of team formation when the team begins to mature, sort out their internal differences, and find solutions to work together. It is considered a period of adjustment.

Number of Stories

Number of Stories refers to the number of User Stories that are delivered as part of a single Sprint. It can be expressed in terms of simple count or weighted count.

Opportunities

Risks that are likely to have a positive impact on the project are referred to as opportunities.

Opportunity Cost

Opportunity cost refers to the value of the next best business option or project that was discarded in favor of the chosen project.

Organizational Deployment Methods

The deployment mechanisms of each organization tend to be different based on industry, target users, and positioning. Depending on the product being delivered, deployment can take place remotely or may involve the physical shipping or transition of an item.

Organizational Resource Matrix

The Organizational Resource Matrix is a hierarchical depiction of a combination of a functional organizational structure and a project organizational structure. Matrix organizations bring together team members for a project from different functional departments such as information technology, finance, marketing, sales, manufacturing, and other departments - and create cross-functional teams.

Paired Comparison

Paired Comparison is a technique where a list of all the User Stories in the Prioritized Product Backlog is prepared. Next, each User Story is taken individually and compared with the other User Stories in the list, one at a time. Each time two User Stories are compared, a decision is made regarding which of the two is more important. Through this process, a prioritized list of User Stories can be generated.

Pareto Analysis

This technique of assessing risk involves ranking risks by magnitude. It helps the Scrum Team address the risks in order of their potential impacts on the project.

PDCA/PDSA cycle

The Plan-Do-Check-Act Cycle—also known as the Deming or Shewhart Cycle—was developed by Dr. W. Edwards Deming, considered the father of modern quality control and Dr. Walter A. Shewhart. Deming later modified Plan-Do-Check-Act to Plan-Do-Study-Act (PDSA) because he felt the term "Study" emphasized analysis rather than simply inspection, as implied by the term "Check." Both Scrum and the Deming/Shewhart/PDCA Cycle are iterative methods that focus on continuous improvement.

Performing stage

The final stage of team formation when the team becomes its most cohesive and operates at its highest level in terms of performance. The members have evolved into an efficient team of peer professionals who are consistently productive.

Personas

Personas are highly detailed fictional characters, representative of the majority of users as well as other stakeholders who may not directly use the end product. Personas are created to identify the needs of the target user base.

Piloting Plan

A Piloting Plan can be used to map out a pilot deployment in detail. The scope and objectives of the deployment, the target deployment user base, a deployment schedule, transition plans, required user preparation, evaluation criteria for the deployment, and other key elements related to the deployment are specified in the Pilot Plan and shared with stakeholders.

Plan and Estimate phase

The Plan and Estimate phase consists of processes related to planning and estimating tasks, which include *Create User Stories*, *Estimate User Stories, Commit User Stories*, *Identify Tasks*, *Estimate Tasks*, and *Create Sprint Backlog*.

Planning for Value

Planning for Value refers to justifying and confirming the project value. The onus for determining how value is created falls on the stakeholders (sponsor, customers, and/or users), while the Scrum Team concentrates on what is to be developed.

Planning Poker

Planning Poker, also called Estimation Poker, is an estimation technique which balances group thinking and individual thinking to estimate relative sizes of User Stories or the effort required to develop them.

Portfolio

A portfolio is a group of related programs, with the objective to deliver business outcomes as defined in the Portfolio Vision Statement. The Prioritized Portfolio Backlog incorporates the Prioritized Program Backlogs for all the programs in the portfolio.

Portfolio Product Owner

The Portfolio Product Owner defines the strategic objectives and priorities for the portfolio.

Portfolio Scrum Master

The Portfolio Scrum Master solves problems, removes impediments, facilitates, and conducts meetings for the portfolio.

Potentially Shippable Deliverables from Projects

Potentially Shippable Deliverables from projects are valuable inputs for coordination at the program or portfolio level. At the end of Sprints in projects, product increments or deliverables are completed. The User Stories included in these increments meet the Definition of Done criteria as well as their respective acceptance criteria.

Prioritization

Prioritizing can be defined as determining the order of things and separating what will be done now, from what can be done later.

Prioritized Product Backlog

The Prioritized Product Backlog is a single requirements document that defines the project scope by providing a prioritized list of features of the product or service to be delivered by the project.

Probability Impact Grid

A grid where Risks are assessed for probability of occurrence and for potential impact on project objectives. Generally, a numerical rating is assigned for both probability and impact independently. The two values are then multiplied to derive a risk severity score, which can be used to prioritize risks.

Probability Trees

Potential events are represented in a diagram with a branch for each possible outcome of the events. The probability of each outcome is indicated on the appropriate branch, and these values can be used to calculate the overall impact of risk occurrence in a project.

Product

The term "product" in the *SBOK™ Guide* may refer to a product, service, or other deliverable that provides value to the customer.

Product Owners Collaboration Plan

The Product Owners Collaboration Plan should define how multiple Product Owners collaborate with the Chief Product Owner.

Prioritized Product Backlog Review Meeting

A Product Backlog Review Meeting (also referred to as a Prioritized Product Backlog Grooming Session) is a formal meeting during the *Groom Prioritized Product Backlog* process, which helps the Scrum Team review and gain consensus about the Prioritized Product Backlog.

Prioritized Program or Portfolio Backlog Review Meetings

At the program or portfolio level, there is representation from each project in the program or from each program in the portfolio. To streamline the meeting, it is generally recommended to have only one representative from each project or program attend at the program or portfolio level.

Product Owner

The Product Owner is the person responsible for maximizing business value for the project. He or she is responsible for articulating customer requirements and maintaining business justification for the project.

Program

A program is a group of related projects, with the objective to deliver business outcomes as defined in the Program Vision Statement. The Prioritized Program Backlog incorporates the Prioritized Product Backlogs for all the projects in the program.

Program and Portfolio Risks

Risks related to a portfolio or program that will also impact projects that are part of the respective portfolio or program.

Program Product Owner

The Program Product Owner defines the strategic objectives and priorities for the program.

Program Scrum Master

The Program Scrum Master solves problems, removes impediments, facilitates, and conducts meetings for the program.

Project

A project is a collaborative enterprise to either create new products or services or to deliver results as defined in the Project Vision Statement. Projects are usually impacted by constraints of time, cost, scope, quality, people and organizational capabilities.

Project Benefits

Project benefits include all measurable improvements in a product, service or result which could be provided through successful completion of a project.

Project Budget

The project budget is a financial document which includes the cost of people, materials, and other related expenses in a project. The project budget is typically signed off by the sponsor(s) to ensure that sufficient funds are available.

Project Charter

A project charter is an official statement of the desired objectives and outcomes of the project. In many organizations, the project charter is the document that officially and formally authorizes the project, providing the team with written authority to begin project work.

Project Costs

Project costs are investment and other development costs for a project

Project Reasoning

Project reasoning includes all factors which support or contribute to the need for the project, whether positive or negative, chosen or not (e.g., inadequate capacity to meet existing and forecasted demand, decrease in customer satisfaction, low profits, and legal requirements).

Project Timescales

Timescales reflect the length or duration of a project. Timescales related to the business case also include the time over which the project's benefits will be realized.

Project Vision Meeting

A Project Vision Meeting is a meeting with the Program Stakeholder(s), Program Product Owner, Program Scrum Master, and Chief Product Owner. It helps identify the business context, business requirements, and stakeholder expectations in order to develop an effective Project Vision Statement.

Project Vision Statement

The key output of the *Create Project Vision* process is a well-structured Project Vision Statement. A good Project Vision explains the business need and what the project is intended to meet rather than how it will meet the need.

Proposed Non-Functional Items for Product Backlog

Non-functional requirements may not be fully defined in the early stages of the project and can surface during the Sprint Review or Retrospect Sprint Meetings. These items should be added to the Prioritized Product Backlog as they are discovered.

Quality

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.

Quality Assurance

Quality assurance refers to the evaluation of processes and standards that govern quality management in a project to ensure that they continue to be relevant. Quality assurance activities are carried out as part of the work.

Quality Control

Quality control refers to the execution of the planned quality activities by the Scrum Team in the process of creating deliverables that are potentially shippable. It also includes learning from each set of completed activities in order to achieve continuous improvement.

Quality Management

Quality management in Scrum enables customers to become aware of any problems in the project early and helps them recognize if a project is going to work for them or not. Quality management in Scrum is facilitated through three interrelated activities:

1. Quality planning
2. Quality control
3. Quality assurance

Quality Planning

Quality Planning refers to identification and definition of the product required from a Sprint and the project along with the Acceptance Criteria, any development methods to be followed, and the key responsibilities of Scrum Team members in regards to quality.

Recommended Scrum Guidance Body Improvements

As a result of planning for the large project, suggestions may be made to revise or enhance the Scrum Guidance Body Recommendations. If the Guidance Body accepts these suggestions, they will be incorporated as updates to the Scrum Guidance Body documentation.

Refactoring

Refactoring is a tool specific to software projects. The aim of this technique is to improve the maintainability of the existing code and make it simpler, more concise, and more flexible. Refactoring means improving the design of the present code without changing how the code behaves. It involves the following:

- Eliminating repetitive and redundant code
- Breaking methods and functions into smaller routines
- Clearly defining variables and method names
- Simplifying the code design
- Making the code easier to understand and modify

Regulations

Regulations include any Federal, Local, State, or industry regulations that the program or portfolio must adhere to. Sometimes, Scrum Guidance Body recommendations may need to be updated to reflect new regulations.

Rejected Deliverables

Rejected Deliverables are the deliverables that do not meet the defined Acceptance Criteria. A list of Rejected Deliverables is maintained and updated after each Sprint Review Meeting with any deliverables that were not accepted.

Rejected Updates to the Scrum Guidance Body Recommendations

Recommended Scrum Guidance Body Improvements may not always be accepted. If the recommended improvement is rejected by the Scrum Guidance Body members, feedback that shares the reason for that rejection is provided to the relevant party.

Relative Prioritization Ranking

Relative Prioritization Ranking is a simple listing of User Stories in order of priority. It is an effective method for determining the desired User Stories for each iteration or release of the product or service.

Relative Sizing/Story Points

In addition to being used for estimating cost, Story Points may also be used for estimating the overall size of a User Story or feature. This approach assigns a story point value based on an overall assessment of the size of a User Story with consideration given to risk, amount of effort required, and level of complexity.

Release Content

This consists of essential information about the deliverables that can assist the Customer Support Team.

Release Notes

Release Notes should include external or market facing shipping criteria for the product to be delivered.

Release Planning Schedule

A Release Planning Schedule is one of the key outputs of the *Conduct Release Planning* process. A Release Planning Schedule states which deliverables are to be released to the customers, along with planned intervals, and dates for releases. There may not be a release scheduled at the end of every Sprint iteration.

Release Planning Sessions

The major objective of Release Planning Sessions is to create a Release Plan Schedule and enable the Scrum Team to have an overview of the releases and delivery schedule for the product they are developing, so that they can align with the expectations of the Product Owner and relevant Stakeholder(s).

Release Preparation Methods

Release preparation methods are the methods used to execute the tasks identified in the Release Readiness Plan in order to get the deliverables ready to be shipped/released.

Release Prioritization Methods

Release Prioritization Methods are used to develop a Release Plan. These methods are industry and organization specific and are usually determined by senior management in an organization.

Release Readiness Sprint

If there is a need for specific tasks to be performed to get ready for a Release and to confirm that the minimum requirements for release have been met, these tasks are performed in a Release Readiness Sprint. A Release Readiness Sprint, if required, is only done once per Release as the last Sprint prior to the Release.

Resolved Issues

In Scrum of Scrums Meetings, Scrum Team members have the opportunity to transparently discuss issues impacting their project. This timely discussion and resolution of issues in the Scrum of Scrums Meeting greatly improves coordination between different Scrum Teams and reduces the need for redesign and rework.

Retrospect Program or Portfolio Meeting

The Retrospect Program or Portfolio Meeting is similar to the Retrospect Project Meeting but is carried out at the program or portfolio level. The major difference is that the frequency of Retrospect Program and Portfolio Meetings is much lower than that of the Retrospect Project Meetings.

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 lessons learned. Often, these lessons lead to the documentation of Agreed Actionable Improvements, to be implemented in future projects.

Retrospect Project Meeting

The Retrospect Project Meeting is a meeting to determine ways in which team collaboration and effectiveness can be improved in future projects. Positives, negatives, and potential opportunities for improvement are also discussed. This meeting is not Time-boxed and may be conducted in person or in a virtual format.

Retrospect Sprint

In this process, the Scrum Master and Scrum Team meet to discuss the lessons learned throughout the Sprint. The lessons learned are documented and can be applied to future Sprints.

Retrospect Sprint Log(s)

The Retrospect Sprint Log is a record of the opinions, discussions, and actionable items raised in a Retrospect Sprint Meeting. The Scrum Master may facilitate creation of this log with inputs form Scrum Core Team members.

Retrospect Sprint Meeting

The Retrospect Sprint Meeting is Time-boxed to 4 hours for a one-month Sprint and conducted as part of the *Retrospect Sprint* process. The length may be scaled up or down relative to the length of the Sprint. During this meeting, the Scrum Team gets together to review and reflect on the previous Sprint in terms of the processes followed, tools employed, collaboration and communication mechanisms, and other aspects relevant to the project.

Return on Investment (ROI)

Return on Investment (ROI), when used for project justification, assesses the expected net income to be gained from a project. It is calculated by deducting the expected costs or investment in a project from its expected revenue and then dividing this (net profit) by the expected costs in order to get a return rate.

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.

Risk Appetite

Risk appetite refers to how much uncertainty a stakeholder or organization is willing to take on.

Risk Assessment

Risk assessment refers to evaluating and estimating identified risks.

Risk Attitude

Essentially, the Risk Attitude of the Stakeholder(s) determines how much risk the Stakeholder(s) consider acceptable. This is a determining factor in when they will decide to take actions to mitigate potential adverse risks.

Risk Averse

Risk Averse is one of the categories of Utility Function. It refers to a Stakeholder being unwilling to accept a risk no matter what the anticipated benefit or opportunity.

Risk Breakdown Structure

In this structure, risks are grouped based on their categories or commonalities. For example, risks may be categorized as financial, technical, or safety related.

Risk Burndown Chart

A chart that depicts cumulative project risk severity over time. The likelihood of the various risks are plotted on top of each other to show cumulative risk on the y-axis. The initial identification and evaluation of risks and the creation of the Risk Burndown Chart are done early in the project.

Risk Checklists

Risk Checklists include key points to be considered while identifying risks, common risks encountered in Scrum projects, or even categories of risks that should be addressed by the team.

Risk Communication

Risk Communication involves communicating the findings from the first four steps of Risk Management to the appropriate Stakeholder(s) and determining their perception regarding the uncertain events.

Risk Identification

Risk Identification is an important step in Risk Management which involves using various techniques to identify all potential risks.

Risk Meeting

Risks can be more easily prioritized by the Product Owner by calling a meeting of the Scrum Core Team and optionally inviting relevant Stakeholders to the meeting.

Risk Mitigation

Risk Mitigation is an important step in Risk Management that involves developing an appropriate strategy to deal with a risk.

Risk Neutral

Risk Neutral is one of the categories of Utility Function that refers to a stakeholder being neither risk averse nor risk seeking; any given decision is not affected by the level of uncertainty of the outcome. When two possible scenarios carry the same level of benefit, the risk neutral stakeholder will not be concerned if one scenario is riskier than the other.

Risk Prioritization

Risk Prioritization is an important step in Risk Management that involves prioritizing risks to be included for specific action in the Prioritized Product Backlog.

Risk Prompt Lists

Risk Prompt Lists are used in stimulating thoughts regarding the source from which risks may originate. Risk Prompt Lists for various industries and project types are available publicly.

Risk Seeking

Risk Seeking is one of the categories of Utility Function that refers to a stakeholder being willing to accept risk even if it delivers a marginal increase in return or benefit to the project.

Risk Threshold

Risk Threshold refers to the level at which a risk is acceptable to the stakeholder's organization. A risk will fall above or below the risk threshold. If it is below, the stakeholder or organization is more likely to accept the risk.

Risk Tolerance

Risk Tolerance indicates the degree, amount, or volume of risk the stakeholders will withstand.

Risk-Based Spike

Risk-Based Spikes are basically experiments that involve research or prototyping to better understand potential risks. In a spike, an intense two to three-day exercise is conducted (preferably at the beginning of a project before the *Develop Epic(s)* or *Create Prioritized Product Backlog* processes) to help the team determine the uncertainties that could affect the project.

Scope

The scope of a project is the total sum of all the product increments and the work required for developing the final product.

Scrum Guidance Body

The Scrum Guidance Body (SGB) is an optional role. It generally consists of a group 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.

Scrum Guidance Body Expertise

Scrum Guidance Body Expertise relates to documented rules and regulations, development guidelines, or standards, and best practices.

Scrum Guidance Body Meetings

The Scrum Guidance Body meets regularly to discuss the potential need for an update of the Scrum Guidance Body Recommendations (e.g., recommended improvements from Retrospectives and other processes, updated regulations, etc.). The frequency of the meetings is decided by the Scrum Guidance Body based on the specific needs of the enterprise.

Scrum Guidance Body Members

The Scrum Guidance Body (SGB) members can include Scrum experts, selected Scrum Masters, Product Owners and team members (on all levels). However, there should be a limit on the number of members that a SGB can have in order to ensure that it remains relevant and does not become prescriptive in nature.

Scrum Master

The Scrum Master is one of the Scrum Core Team roles. He or she facilitates creation of the project's deliverables, manages risks, changes, and impediments during the *Conduct Daily Standup, Retrospect Sprint,* and other Scrum processes.

Scrum of Scrums Meeting

A Scrum of Scrums Meeting is an important element when scaling Scrum to large projects. Typically, there is one representative in the meeting from each Scrum Team—usually the Scrum Master—but it is also common for anyone from the Scrum Team to attend the meeting if required. This meeting is usually facilitated by the Chief Scrum Master and is intended to focus on areas of coordination and integration between the different Scrum Teams.

Scrum Team

The Scrum Team is one the Scrum Core Team roles. The Scrum Team works on creating the deliverables of the project and contributes to realizing business value for all stakeholders and the project.

Scrum Teams Collaboration Plan

The Scrum Teams Collaboration Plan defines how the numerous Scrum Teams collaborate with each other to provide highest value in the shortest possible time.

Scrum Team Lessons Learned

The self-organizing and empowered Scrum Team is expected to learn from mistakes made during a Sprint and these lessons learned help the teams improve their performance in future Sprints.

Scrum Team Representatives

A representative nominated by the team to represent them in the Scrum of Scrums (SoS) Meetings based on who can best fulfill the role depending on current issues and circumstances.

Scrumboard

Scrumboard is a tool used by the Scrum Team to plan and track progress during each Sprint. The Scrumboard contains four columns to indicate the progress of the estimated tasks for the Sprint: a To Do column for tasks not yet started, an In Progress column for the tasks started but not yet completed, a Testing column for tasks completed but in the process of being tested, and a Done column for the tasks that have been completed and successfully tested.

Self-organization

Scrum believes that employees are self-motivated and seek to accept greater responsibility. Hence, they deliver much greater value when self-organized.

Servant Leader

Servant leaders employ listening, empathy, commitment, and insight while sharing power and authority with team members. Servant leaders are stewards who achieve results by focusing on the needs of the team. This style is the embodiment of the Scrum Master role.

Shared Resources

Shared resources can include people, environment, and equipment that are needed by all or some of the Scrum Teams working on the project. In a large project, the shared resources may be limited and are needed by all or some of the Scrum Teams at the same time.

Ship Deliverables

In this process, Accepted Deliverables are delivered or transitioned to the relevant Stakeholder(s). A formal Working Deliverables Agreement documents the successful completion of the Sprint.

Simple Schemes

Simple Schemes involve labeling items as Priority “1”, “2”, “3” or “High”, “Medium” and “Low” and so on. Although this is a simple and straightforward approach, it can become problematic because there is often a tendency to label everything as Priority “1” or “High”.

Skills Requirement Matrix

The skills requirement matrix, also known as a competency framework, is used to assess skill gaps and training requirements for team members. A skills matrix maps the skills, capabilities, and interest level of team members in using those skills and capabilities on a project. Using this matrix, the organization can assess any skill gaps in team members and identify the employees who will need further training in a particular area or competency.

Speed Boat

Speed Boat is a technique that can be used to conduct the Retrospect Sprint Meeting. Team members play the role of the crew on a Speed Boat. The boat must reach an island, which is symbolic of the Project Vision. Sticky notes are used by the attendees to record engines and anchors. Engines are things which help them reach the island, while anchors are things that are hindering them from reaching the island. This exercise is time-boxed to a few minutes.

Sponsor

The sponsor is the individual or the organization that provides resources and support for the project. The sponsor is also the stakeholder to whom everyone is accountable in the end.

Sprint

A Sprint is a time-boxed iteration of one to six weeks in duration during which the Scrum Team works on and creates the Sprint deliverables.

Sprint Backlog

Sprint Backlog is a list of the tasks to be executed by the Scrum Team in the upcoming Sprint.

Sprint Burndown Chart

Sprint Burndown Chart is a graph that depicts the amount of work remaining in the ongoing Sprint.

Sprint Deliverables

Sprint Deliverables refer to product increments or deliverables that are completed at the end of each Sprint.

Sprint Planning Meeting

Sprint Planning Meeting is conducted at the beginning of a Sprint as part of the *Create Sprint Backlog* process. It is Time-boxed to eight hours for a one-month Sprint and is divided into two parts - Objective Definition and Task Estimation.

Sprint Review Meeting

The Sprint Review Meeting is time-boxed to four hours for a one-month Sprint and can be scaled according to the length of the Sprint. During the Sprint Review Meeting, the Scrum Team presents the deliverables of the current Sprint to the Product Owner, who may accept or reject the deliverables.

Sprint Tracking Tools

Sprint Tracking Tools are used to track the progress of a Sprint and to know where the Scrum Team stands in terms of completing the tasks in the Sprint Backlog. A variety of tools can be used to track the work in a Sprint, but one of the most common is a Scrumboard, also known as a task board or progress chart.

Sprint Velocity

Sprint Velocity is the rate at which the team can complete the work in a Sprint. It is usually expressed in the same units as those used for estimating, normally story points or ideal time.

Stakeholder(s)

Stakeholder(s) is a collective term that includes customers, users, and sponsor who frequently interface with the Product Owner, Scrum Master and Scrum Team to provide inputs and facilitate creation of the project's product, service, or other results.

Stakeholder Analysis

A standard stakeholder analysis is used to identify the stakeholders on program and portfolio levels. Further details related to program or portfolio stakeholders may be identified as personas in the Create and Groom Program or Portfolio Backlog process.

Storming Stage

The second stage of team formation where the team begins trying to accomplish the work. However, power struggles may occur and there is often chaos or confusion among team members.

Story Mapping

Story Mapping is a technique to provide a visual outline of the product and its key components. Story Mapping, first formulated by Jeff Patton (2005), is commonly used to illustrate product roadmaps. Story maps depict the sequence of product development iterations and map out which features will be included in the first, second, third, and subsequent releases.

Sustainable Pace

Sustainable Pace is the pace at which the team can work and comfortably maintain. It translates to increased employee satisfaction, stability, and increased estimation accuracy, all of which ultimately leads to increased customer satisfaction.

SWOT Analysis

SWOT is a structured approach to project planning that helps evaluate the strengths, weaknesses, opportunities, and threats related to a project. This type of analysis helps identify both the internal and the external factors that could impact the project.

Target Customers for Release

Not every release will target all stakeholders or users. The Stakeholders may choose to limit certain releases to a subset of users. The Release Plan specifies the Target Customers for the Release.

Task Estimation Workshop

Task Estimation Workshops enable the Scrum Team to estimate the effort required to complete a task or set of tasks and to estimate the people effort and other resources required to carry out the tasks within a given Sprint.

Task List

This is a comprehensive list that contains all the tasks to which the Scrum Team has committed to for the current Sprint. It contains descriptions of each task.

Task Planning Meeting

In a Task Planning Meeting, the Scrum Team gets together to plan the work to be done in the Sprint and the team reviews the committed User Stories at the top of the Prioritized Product Backlog. To help ensure that the group stays on topic, this meeting should be Time-boxed, with the standard length limited to two hours per week of Sprint duration.

Task-Oriented Leader

Task-Oriented Leaders enforce task completion and adherence to deadlines.

Team Building Plan

Since a Scrum Team is cross-functional, each member needs to participate actively in all aspects of the project. The Scrum Master should identify potential issues that could crop up with team members and try to address them diligently in the Team Building Plan in order to maintain an effective team.

Team Calendar

A Team Calendar contains information regarding availability of team members including information related to employee vacations, leaves, important events, and holidays.

Team Expertise

Team Expertise refers to the expertise of the Scrum Team members to understand the User Stories and Tasks in the Sprint Backlog in order to create the final deliverables. Team Expertise is used to assess the inputs needed to execute the planned work of the project.

Team Specialization

In a large project, Team Specialization may be required. There are three dimensions of Team Specialization. The first dimension is the need for accomplishing specific tasks. The second dimension is the need for special skills of single team members. The third dimension is that there may be limitations in team flexibility.

Technical Debt

Technical Debt (also referred to as design debt or code debt) refers to the work that teams prioritize lower, omit, or do not complete as they work towards creating the primary deliverables associated with the project's product. Technical Debt accrues and must be paid in the future.

Theory X

Theory X leaders assume that employees are inherently unmotivated and will avoid work if possible, warranting an authoritarian style of management.

Theory Y

Theory Y leaders assume that employees are self-motivated and seek to accept greater responsibility. Theory Y involves a more participative management style.

Threats

Threats are risks that could affect the project in a negative manner.

Three Daily Questions

Three Daily Questions used in Daily Standup Meetings which are facilitated by the Scrum Master, where each Scrum Team member provides information in the form of answers to three specific questions:

- What have I done since the last meeting?
- What do I plan to do before the next meeting?
- What impediments or obstacles (if any) am I currently facing?

Time-boxing

Time-boxing refers to setting short periods of time for work to be done. If the work undertaken remains incomplete at the end of the Time-box, it is moved into a subsequent Time-box. Time-boxes provide the structure needed for Scrum projects, which have an element of uncertainty, are dynamic in nature, and are prone to frequent changes.

Transparency

Transparency allows all facets of any Scrum process to be observed by anyone. Sharing all information leads to a high trust environment.

Unapproved Change Requests

Request for changes are usually submitted as Change Requests. Change Requests remain unapproved until they get formally approved.

Updated Program Product Backlog

A Program Product Backlog that undergoes periodic grooming to incorporate changes and new requirements.

Updated Scrum Guidance Body Membership

As a result of assessing the Scrum Guidance Body membership, new members may be included in the Scrum Guidance Body and existing members may be removed or leave the Scrum Guidance Body.

Updated Implementation Deadlines for Projects

Implementation deadlines for projects may be updated to reflect the impact of new or changed User Stories that need to modify or introduce new requirements.

Updated Prioritized Program or Portfolio Backlog

The Prioritized Program or Portfolio Backlog may be updated with new User Stories, new change requests, new identified risks, updated User Stories, or reprioritization of existing User Stories.

User

Users are the individuals or the organization that directly uses the project's product, service, or other results. Like customers, for any organization, there can be both internal and external users. In some cases, customers and users may be the same.

User Group Meetings

User Group Meetings involve relevant Stakeholder(s), primarily users or customers of the product. They provide the Scrum Core Team with first-hand information about user expectations. This helps in formulating the Acceptance Criteria for the product and provides valuable insights for developing Epics.

User Stories

User Stories adhere to a specific, predefined structure and are a simplistic way of documenting the requirements and desired end-user functionality. The requirements expressed in User Stories are short, simple, and easy-to-understand statements resulting in enhanced communication among the stakeholders and better estimations by the team.

User Story Acceptance Criteria

Every User Story has associated Acceptance Criteria. User Stories are subjective, so the Acceptance Criteria provide the objectivity required for the User Story to be considered as Done or not Done during the Sprint Review providing clarity to the team on what is expected of a User Story.

User Story Workshops

User Story Workshops are held as part of the *Develop Epic(s)* process. The Scrum Master facilitates these sessions. The entire Scrum Core Team is involved and at times it is desirable to include other Stakeholder(s).

User Story Writing Expertise

The Product Owner, based on his or her interaction with the stakeholders, business knowledge and expertise, and inputs from the team, develops User Stories that form the initial Prioritized Product Backlog for the project.

Utility Function

Utility Function is a model used for measuring stakeholder risk preference or attitude toward risk. It defines the Stakeholder(s)' level or willingness to accept risk.

Value Stream Mapping

Value Stream Mapping uses flowcharts to illustrate the flow of information needed to complete a process and may be used to streamline a process by helping to determine non-value-adding elements.

Vendor

Vendors include external individuals or organizations that provide products and services that are not within the core competencies of the project organization.

Voice of the Customer (VOC)

The Voice of the Customer (VOC) can be referred to as the explicit and implicit requirements of the customer, which must be understood prior to the designing of a product or service. The Product Owner represents the Voice of the Customer.

War Room

War Room is the commonly used term to describe the location where all Scrum Team members working are located. Normally, it is designed in such a way that team members can move around freely, work, and communicate easily because they are located in close proximity to each other.

Wideband Delphi Technique

Wideband Delphi is a group-based estimation technique for determining how much work is involved and how long it will take to complete. Individuals within a team anonymously provide estimations for each feature and the initial estimates are then plotted on a chart. The team then discusses the factors that influenced their estimates and proceed to a second round of estimation. This process is repeated until the estimates of individuals are close to each other and a consensus for the final estimate can be reached.

Working Deliverables

This output is the final shippable deliverable for which the project was sanctioned.

Working Deliverables Agreement

Deliverables that meet the Acceptance Criteria receive formal business sign-off and approval by the customer or the sponsor.

INDEX

1

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

The essential guide to successfully deliver projects using Scrum

The *SBOK™ Guide* was developed as a means to create a necessary guide for organizations and professionals who want to implement Scrum, as well as those already doing so who want to make needed improvements to their existing 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 delivery practitioners have been considered in its development. The focus of Scrum on value-driven delivery helps Scrum Teams deliver results as early in the project as possible, thus improving Return on Investment for those companies which use Scrum as their preferred project delivery framework. Moreover, managing changes is easy through the use of short, iterative product development cycles and frequent interaction between the customers and the Scrum Teams.

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 individuals with no prior experience or knowledge of Scrum or other project delivery method. The first chapter describes the purpose and framework of the *SBOK™ Guide* and provides an introduction to the key concepts of Scrum and a summary of the Scrum principles, aspects, and 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 Scrum aspects that must be addressed throughout any project—organization, business justification, quality, change, and risk. Chapters 8 through 12 cover the 19 fundamental Scrum processes involved in carrying out a Scrum project. These processes are part of the 5 Scrum phases of Initiate; Plan and Estimate; Implement, Review and Retrospect; and Release. Details about the associated inputs and outputs of each process, as well as the various tools that may be used in each are described.

This third edition of the *SBOK™ Guide* adds to the collective knowledge of the Scrum framework with expanded content related to scaling Scrum for large projects, and scaling Scrum for the Enterprise, covered in Chapters 13 and 14 respectively.

Although the *SBOK™ Guide* is a very comprehensive reference book for Scrum, its contents are organized for easy reference and enjoyable reading, irrespective of the reader's prior knowledge of Scrum.

SCRUMstudy is a brand of VMEdu, Inc.