

A Guide to the

# SCRUM BODY OF KNOWLEDGE

(SBOK® Guide)

---

## APPENDIX A. ADDITIONAL LEARNING

**A Comprehensive Guide to Implementing  
and Scaling Scrum, with Practical Examples**

(Includes insights into how Artificial Intelligence can enhance Scrum processes)

## APPENDIX A. OVERVIEW OF AGILE

### A.1 Introduction

This appendix intends to familiarize readers with the concept of Agile development and the various Agile methodologies.

The following sections are included:

**A.2 Overview**—This section discusses the definition of and the factors behind the rise of Agile.

**A.3 Agile Manifesto**—This section presents *The Agile Manifesto*, its principles, and *The Declaration of Interdependence* to provide the historical context of Agile.

**A.4 Agile Methods**—This section provides a brief overview of specific Agile methodologies including:

- Lean Kanban
- Extreme Programming
- Crystal Methods
- Dynamic Systems Development Methods
- Feature Driven Development
- Test Driven Development
- Adaptive Software Development
- Agile Unified Process
- Domain-Driven Design

## A.2 Overview

The term “Agile” refers to being able to move or respond quickly and easily; being nimble. In any kind of management discipline, Agile as a quality would therefore be a valid aim. Agile project management specifically involves being adaptive during the creation of a product, service, or other deliverable.

It is important to understand that while Agile development methods are highly adaptive, it is also necessary to incorporate stability in their adaptive processes.

### A.2.1 The Rise of Agile

Rapid changes in technology, market demands, and expectations have resulted in increased challenges to developing products and services using traditional project management models. This paved the way for the conceptualization and implementation of Agile methods and values in many organizations. Agile development models addressed the shortcomings associated with traditional project management models in meeting the ever-growing environmental demands and expectations that organizations were facing. Since traditional project management models emphasize extensive upfront planning and conforming to the plan once it is baselined, such models were not successful in meeting the reality of frequent environmental changes.

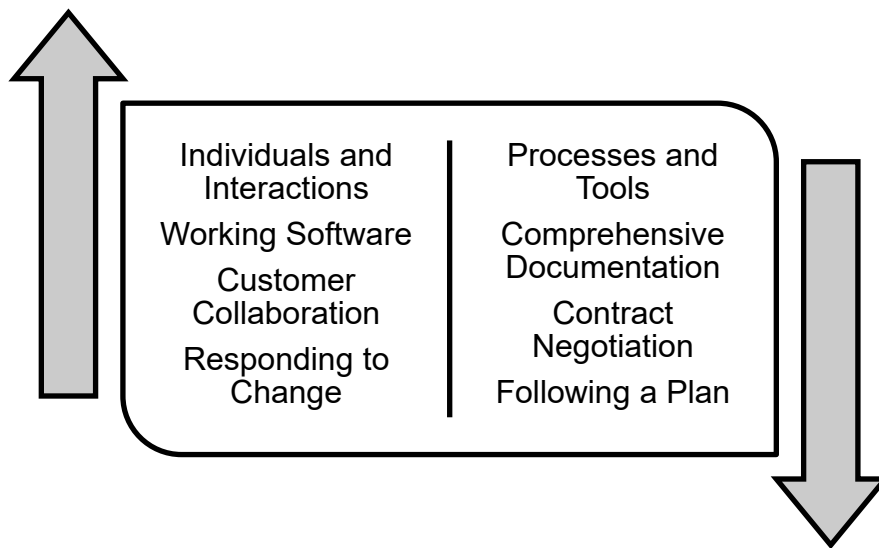
Agile relies on adaptive planning and iterative development and delivery. It focuses primarily on the value of people in getting the job done effectively. Though adaptive and incremental methodologies have existed since the 1950s, only methodologies that conform to *The Agile Manifesto* are regarded as truly “Agile”.

## A.3 The Agile Manifesto

In February 2001, a group of seventeen computer gurus, software developers, and managers held a retreat to discuss lightweight software development methods. They formed the *Agile Alliance* and the discussions at those meetings later resulted in a *Manifesto for Agile Software Development*. The Manifesto was authored by Fowler and Highsmith (2001) and then signed by all participants to establish the basic guidelines for any Agile method.

The purpose of *The Agile Manifesto* was laid out as follows:

*We are uncovering better ways of developing  
software by doing it and helping others do it.  
Through this work we have come to value:*



*That is, while there is value in the items on  
the right, we value the items on the left more.*

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham  
Martin Fowler

James Grenning  
Jim Highsmith  
Andrew Hunt  
Ron Jeffries  
Jon Kern  
Brian Marick

Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dave Thomas

Permission to copy provided by the above authors by notice on <http://agilemanifesto.org/>.

The four trade-offs emphasized by The Agile Manifesto are elaborated as follows:

**1. Individuals and interactions over processes and tools**

Although processes and tools help in successfully completing a project, it is the individuals who undertake, participate in, implement a project, and determine which processes and tools to use. The key actors in any project are therefore individuals, so the emphasis should be on them and their interactions, rather than complicated processes and tools.

**2. Working software over comprehensive documentation**

While documentation is necessary and useful for any project, many teams focus on collecting and recording qualitative and quantitative descriptions of deliverables, when the real value delivered to the customer is primarily in the form of working software. Therefore, the Agile focus is on delivering working software in increments throughout the product lifecycle rather than detailed documentation.

**3. Customer collaboration over contract negotiation**

Traditionally, customers have been seen as outside players who are involved at the start and end of the product lifecycle and whose relationships were based on contracts and their fulfillment. Agile believes in a shared value approach in which customers are seen as collaborators. The development team and customers work together to evolve and develop the product.

**4. Responding to change over following a plan**

In the current market in which customer requirements, available technologies, and business patterns are constantly changing, it is essential to approach product development in an adaptive manner that enables change incorporation and fast product development lifecycles rather than emphasizing following plans formed with potentially outdated data.

### A.3.1 Principles of the Agile Manifesto

The twelve principles of the Agile Manifesto by Fowler and Highsmith (2001) are:

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

### A.3.2 Declaration of Interdependence

The Agile project management *Declaration of Interdependence* was written in early 2005 by a group of fifteen project leaders as a supplement to *The Agile Manifesto*. It enumerates six management values needed to reinforce an Agile development mentality, particularly when managing complex, uncertain projects.

The declaration highlights that project teams, customers, and other business stakeholders are interdependent and connected and must recognize this to be successful. The values themselves are also interdependent.

We ...

**increase return on investment** by making continuous flow of value our focus.

**deliver reliable results** by engaging customers in frequent interactions and shared ownership.

**expect uncertainty** and manage for it through iterations, anticipation, and adaptation.

**unleash creativity and innovation** by recognizing that individuals are the ultimate source of value and creating an environment where they can make a difference.

**boost performance** through group accountability for results and shared responsibility for team effectiveness.

**improve effectiveness and reliability** through situationally specific strategies, processes, and practices.

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., et al. 2005

## A.4 Agile Methods

A number of Agile methodologies originated and gained traction in the 1990's and the early 2000's. While they differ in a variety of aspects, their commonality stems from their adherence to *The Agile Manifesto*.

The following Agile methods are briefly discussed below:

1. Lean Kanban
2. Extreme Programming (XP)
3. Crystal Methods
4. Dynamic Systems Development Methods (DSDM)
5. Feature Driven Development (FDD)
6. Test Driven Development (TDD)
7. Adaptive Software Development (ASD)
8. Agile Unified Process (AUP)
9. Domain-Driven Design (DDD)

### A.4.1 Lean Kanban

The Lean concept optimizes an organization's system to produce valuable results based on its resources, needs, and alternatives while reducing waste. Waste could be from building the wrong thing, failure to learn, or practices that impede the process. Because these factors are dynamic in nature, a lean organization evaluates its entire system and continuously fine tunes its processes. The foundation of Lean is that the reduction of the length of each cycle (i.e., an iteration) leads to an increase in productivity by reducing delays, aids in error detection at an early stage, and consequently reduces the total amount of effort required to finish a task. Lean software principles have been successfully applied to software development.

Kanban means a “signboard” or “billboard” and it espouses the use of visual aids to assist and track production. The concept was introduced by Taiichi Ohno considered to be the father of the Toyota Production Systems (TPS). The use of visual aids is effective and has become common practice. Examples include task cards, Scrumboards, and Burndown Charts. These methods gained attention due to their practice at Toyota, a leader in process management. Lean Kanban integrates the use of the visualization methods as prescribed by Kanban along with the principles of Lean creating a visual incremental evolutionary process management system.

### A.4.2 Extreme Programming

Extreme Programming (XP), which originated in Chrysler Corporation, gained traction in the 1990's. XP makes it possible to keep the cost of changing software from rising radically over time. The key attributes of XP include incremental development, flexible scheduling, automated test codes, verbal communication, ever-evolving design, close collaboration, and tying in the long- and short-term drives of all those involved.



XP values communication, feedback, simplicity, and courage. The different roles in the XP approach include customer, developer, tracker, and coach. It prescribes various coding, developer, and business practices as well as events and artifacts to achieve effective and efficient development. XP has been extensively adopted due to its well-defined engineering practices.

### **A.4.3 Crystal Methods**

Alistair Cockburn introduced the Crystal methodologies of software development in the early 1990s. Crystal methods are intended to be people-centric, lightweight, and easy to adapt. Because people are primary, the developmental processes and tools are not fixed but are adjusted to the specific requirements and characteristics of the project. The color spectrum is used to decide on the variant for a project. Factors such as comfort, discretionary money, essential money, and life play a vital role in determining the “weight” of the methodology, which is represented in various colors of the spectrum. The Crystal family is divided into Crystal Clear, Crystal Yellow, Crystal Orange, Crystal Orange Web, Crystal Red, Crystal Maroon, Crystal Diamond, and Crystal Sapphire.

All Crystal methods have four roles—executive sponsor, lead designer, developers, and experienced users. Crystal Methods recommend various strategies and techniques to achieve agility. A Crystal project cycle consists of chartering, delivery cycle, and wrap-up.

### **A.4.4 Dynamic Systems Development Methods (DSDM)**

The Dynamic Systems Development Methods (DSDM) framework was initially published in 1995 and is administered by the DSDM Consortium. DSDM sets quality and effort in terms of cost and time at the outset and adjusts the project deliverables to meet set criteria by prioritizing the deliverables into “Must have,” “Should have,” “Could have,” and “Won’t have” categories (using the MoSCoW prioritization technique). DSDM is a system-oriented method with six distinct phases—Pre-project; Feasibility; Foundations; Exploration and Engineering; Deployment; and Benefit Assessment.

### **A.4.5 Feature Driven Development (FDD)**

Feature Driven Development (FDD) was introduced by Jeff De Luca in 1997 and operates on the principle of completing a project by breaking it down into small, client-valued functions that can be delivered in less than two weeks’ time. FDD has two core principles—software development is a human activity and software development is a client-valued functionality.

FDD defines six major roles—Project Manager, Chief Architect, Development Manager, Chief Programmers, Class Owners, and Domain Experts with a number of supporting roles. The FDD process is iterative and consists of developing an overall model, building a feature list, and then planning, designing, and building by feature.

### **A.4.6 Test Driven Development (TDD)**

Sometimes known as Test-First Development, Test Driven Development was introduced by Kent Beck, one of the creators of Extreme Programming (XP). Test Driven Development is a software development method that involves writing automated test code first and developing the least amount of code necessary to pass that test later. The entire project is broken down into small, client-valued features that need to be developed in the shortest possible development cycle. Based on clients' requirements and specifications, tests are written. The tests designed in the above stage are used to design and write the production code.

TDD can be categorized into two levels: Acceptance TDD (ATDD) requiring a distinct acceptance test and Developer TDD (DTDD) involving writing a single developer test. TDD has become popular because of the numerous advantages it offers like rapid and reliable results, constant feedback, and reduced debugging time.

### **A.4.7 Adaptive Software Development (ASD)**

Adaptive Software Development (ASD) grew out of the rapid application development work by Jim Highsmith and Sam Bayer. The highlights of ASD are constant adaptation of processes to the work at hand, provision of solutions to problems surfacing in large projects, and iterative, incremental development with continuous prototyping.

Being a risk-driven and a change-tolerant development approach, ASD believes a plan cannot admit uncertainties and risks as this indicates a flawed and failed plan. ASD is feature-based and target-driven. The first phase of development in ASD is Speculate (as opposed to Planning) followed by the Collaborate and Learn phases.

### **A.4.8 Agile Unified Process (AUP)**

Agile Unified Process (AUP) evolved from IBM's Rational Unified Process. Developed by Scott Ambler, AUP combines industry-tried-and-tested Agile techniques such as Test-Driven Development (TDD), Agile Modeling, Agile change management, and database refactoring, to deliver a working product of the best quality.

AUP models its processes and techniques on the values of Simplicity, Agility, Customizability, Self-organization, Independence of tools, and focus on high-value activities. The AUP principles and values are put into action in the phases of Inception, Elaboration, Construction, and Transition.

### A.4.9 Domain-Driven Design (DDD)

Domain-driven design is an Agile development approach meant for handling complex designs with implementation linked to an evolving model. It was conceptualized by Eric Evans in 2004 and revolves around the design of a core domain. “Domain” is defined as an area of activity to which the user applies a program or functionality.

Many such areas are batched, and a model is designed. The model consists of a system of abstractions that can be used to design the overall project and solve the problems related to the batched domains. The core values of DDD include domain-oriented, model-driven design, ubiquitous language, and a bounded context.

In DDD, ubiquitous language is established, and the domain is modeled. Then design, development, and testing follow. Refining and refactoring of the domain model is done until it is satisfactory.

## 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 4<sup>th</sup> Edition*.

### B.1 Lead Author

Tridibesh Satpathy

### B.2 Coauthors and Subject Matter Expert Committee

Winfried Hackmann

Gaynell Malone

Ruth Kim

Buddy Peacock

Deepak Ramaswamy

Quincy D. Jordan

J. Drew Nations

Karen Lyncook

Jaimie M. Rush

Elizabeth Lynne Warren

Gaurav Garg

Dipaka Patra

Ahmed Touseefullah Siddiqui

Nikhil Kumar

### B.3 Contributors and Reviewers

Abdelnaser Dwaikat, Btech, Mtech

Abhijit Daayma, MBA, SMC

Aimee Norman, SMC

Alec Vasquez, SMC

Alejandro Córdova, MBA, PMP, PMI-RMP, SCT

Angela Mascarenas, BSC, SMC, SPOC  
Aniruddha Banerjee, SMC, SFC  
Antonio Marcias, SPOC, SFC  
Anu Ravi, BE, SPS, PMP, ITIL, PSM1  
Arturo Velazquez, MSC, SMC, SFC  
Barbara Siefken, MBA, PMP, ITIL, CSM  
Bobbie Green, MBA  
Brian Rubin, SPOC  
Bryan Lee Perez, MS, PMP, CSM, SSGB, SMC  
Carlos Acuña, MBA, PMP, PgMP, CBAP, RMP, SCT  
Charles J. Quansah, MSIT, CCNA, PMP, ITIL  
Charles Letterman, SMC  
Chrys Thorsen, CCSI, MCSE, CISSP  
Corey Bailey, PMP, ITIL, SMC, SAMC  
Corky Henderson, MBA, PMP, SMC, SSGB  
Cristian Mauricio Avila Patarroyo, SPOC  
Damien Lee, PMP, ACP, SMC, SPOC  
David Soden, SMC  
Deepa K SMC, SPOC, SFC  
Derik Stalls, PMP, ITIL, SPOC  
Dusan Kamenov, PMP, PRINCE2, CSM, SFC  
Efetobore Adebayo Omadevuae, MBA, ITIL, SAFe-SA, SMC  
Enrique Vanegas, SMC, SPOC  
Ernesto Ibarra, SMC, SPOC  
Frances Mary Jo Tessler, PMP  
Frank Quinteros, MBA, PMP, PMI-ACP  
Gabriel Joseph, MSC, SMC, SAMC, SPOC  
Ganesh Watve, MBA, PMP, SCT  
George Hanosh, BSC, SMC, SPOC  
Gerald Varghese, SMC, SPOC  
Girish Kulkarni, MSc, ITIL, PRINCE2, CSM  
Ian Glenister, BA, PRINCE2, PMP, SMC, SPOC  
Ignacio Navarro Zaragoza, PMP, ITIL, LEAN IT, SCT  
Inder Mohan Singh, SMC, SFC  
Isaiah Rajan, PMP, MBA, SMC, SFC  
James Cajuste, SMC, SFC  
James McHarry, SMC  
James Pruitt, ITIL, SMC, SPOC  
James Rafferty, SPOC  
Jared Smuli, SMC  
Javier González, MBA, PMI-ACP, ITIL, MCP, SCT  
Jeff Blitstein, MBA, PMP, ACP, SPOC  
Jim Huckin, PMP, SMC  
Jo Pereira, SMC

Joe Schofield, SAFe- SA, LSSBB, CSQA, CMMI, SCT  
Jose Antonio Pineda Mora, MBA, PMP, ITIL, SCT  
Jose Nunez, PhD, PE, PMP  
Joshua Adelankun, SMC, SPOC  
Juan Alberto Marques Rodriguez, SPOC  
Juan Carlos Linares, PMP, SMC  
Katherine Ricker, PMP, SFC, SPOC  
Kuljeet Singh Sarna, MBA, PRINCE2, PMI-ACP, CISSP, SCT  
Lachlan McGurk, PMP, ITIL, SSBB, SMC  
Lennon Burhannan, SMC, SAMC, SPOC, SFC, SSMC  
Lucy Vorpahl, BBA, SMC, SPOC, SFC  
Madhuresh Kumar Mishra, PMP, PRINCE2, ITIL, SMC, SPOC  
Magaline D. Harvey, MBA, PMP, SMC  
Mariela Laborde, MBA, SMC, SPOC, SFC  
Meena Elangovan, BE, PMP, SSGB, PSM  
Melissa Lauro, MA, SMC, SAMC  
Melvin Wofford Jr., BSc, PMP  
Michael Harmon MBA, ITIL, SMC, SFC  
Michael Rauch, MBA, PMP, ACP, PSM1  
Michael Sanchez, SPOC  
Michael W Madigan, SMC, SAMC, SPOC, SFC  
Michelle Wilkinson, MBA, PMP, SMC  
Mike Tomaszewski, PgMP, PMP, MBA  
Mimi LaRaue, PMP, PMI-ACP  
Miriam Kirkaldie, SMC  
Mitch Malloy, SPC, RTE, SMC, SPOC  
Monica Strazzante, PMP, SMC  
Morris Feigel, PMP, PRINCE2, SPC, ITIL, PSPO1  
Muminul Haque, MSC, SMC, SFC  
Nadra Rafee, PMP, SAFe-SA, SMC  
Neha Mishra, BBA, MBA,  
Nichole Thompson, MBA, SPC, SMC  
Nikhil Bhargava, BE, MBA, SMC, SPOC  
Obi Nwaojigba, MBA, PMP, ITIL, SPOC  
Olatunde Badmus, MBA, PMP, SMC  
Olumide Idowu, PMP, ACPC, SMC  
Oscar Esquivel, BSC, SMC, SFC  
Paul de Cunzo, SMC, SPOC  
Prof. Dr. Akram Hassan, MBA, PMP, RMP, SFC  
Ranjit Majumda, SMC, SPOC, SFC  
Raul Caban, ITIL, SMC, SPOC  
Ravi Kumar Kalose, MCA, PGDBA, PMP, SCT  
Ravneet Kaur, MBA, SMC, RTO  
Richard Mather, MSc, PG

Rima Vyas, SMC  
Robert Lamb, PMP, MCCT, MCSE  
Romil Desai MBA, PMP, LSSBB, SMC, SFC  
Ron Villmow, MSc, MCT  
Sandra A. Strech, PMP  
Sandy (Sanjukta) Banerjee, SMC  
Santosh Heroorker, MS, SMC, SFC  
Saurabh Gupta, BE, SAFe, AINS  
Sean McVeigh, SMC  
Seun Odunlami, LPM, SAFe-SA, SMC, SFC  
Sheri Palmer, PgMP, PMP, SMC, SFC  
Sheryl Cattrell, SPOC, SFC  
Simon Robertson, PMP, MSP, SMC, SAMC, SCT  
Sohini Banerjee, BA, SMC  
Sourabh Sharma, SMC  
Srikanth PV, MBA, CMA, PMP, SSGB, SCT  
Srinivas Reddy Kandi, MBA, MCA, IOT  
Steve Versteer, SMC  
Sudheer Vankadara, MBA, SDC, SMC, SAMC, SPOC  
Sunil Krishnan, MBA, SMC  
Syed Ashraf, BSc, MCA, PRINCE2  
Thomas Nelson Woltz, SMC, SFC  
Tommie L. Sherrill, MBA, PMP, SMC  
Tracey Branch, SMC  
Tushar Purohit, PRINCE2, ITIL, SSGB, SMC  
Vicente Manuel Guerra Hernández, SDC, SMC  
Vince Belanger, BASc, PMP  
Vinod Kumar, MCA  
Yogaraj Mudalgi, BA, SMC  
Yvonne Van Horn, SPOC

## APPENDIX C. FOURTH EDITION UPDATES

This appendix provides a summary of updates implemented in the *SBOK® Guide—Fourth Edition* as compared to previous editions.

### C.1 Summary of Changes

The scope of updates made for the *SBOK® Guide—Fourth 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.
- ‘Retrospect Project’ process has been changed to ‘Retrospect Release’ process
- Non-core Role ‘Stakeholders’ has been changed to ‘Business Stakeholders’
- ‘Servant Leadership’ has been changed to ‘Supporting Leadership’
- ‘Create Sprint Backlog’ process has been changed to ‘Update Sprint Backlog’ process
- ‘Groom Prioritized Product Backlog’ process has been changed to ‘Refine Prioritized Product Backlog’ process
- ‘Create Program or Portfolio Components’ process has been changed to ‘Create / Update Program or Portfolio Components’ process
- Tool ‘Scrum Team Selection’ in ‘Form Scrum Team’ process has been changed to ‘Scrum Team Selection Criteria’
- Optional output ‘High-Level Estimates for Epics’ has been added to the process ‘Create Prioritized Product Backlog’
- Optional input ‘Pre-Existing Estimates for User Stories’ has been added to the process ‘Estimate User Stories’
- Optional input ‘Pre-Existing Estimates for Tasks’ has been added to the process ‘Estimate Tasks’

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 Fourth Edition Updates by Chapter

Chapter	Key Changes Made
1	<ul style="list-style-type: none"> <li>• Improved consistency and clarity.</li> <li>• Updated section 1.1.1 to accurately reflect the history of Scrum.</li> <li>• Added reference to two new certifications, SSMC™ and SSPOC™ (section 1.3).</li> <li>• Figure 1-2: SBOK® Guide Framework is updated.</li> <li>• Non-core role 'Stakeholders' is changed to 'Business Stakeholders.'</li> <li>• Figure 1-4: Organization in Scrum is updated.</li> <li>• Table 1-1: Summary of Scrum Fundamental Processes is updated to reflect the changes made in the names of Scrum processes.</li> <li>• Table 1-2: Scrum Meetings and Processes is added to capture all the key Scrum meetings</li> <li>• Updated Scrum Processes (section 1.4.4) to reflect new process names for Plan and Estimate phase (see chapter 9). Added the inputs, tools, and outputs discussed in chapter 13 Scaling Scrum for Large Projects. Updated processes discussed in chapter 14 Scaling Scrum for the Enterprise.</li> <li>• Added new section 1.4.4.6 Scrum Meetings or Ceremonies.</li> <li>• Updated sections 1.4.4.7 and 1.4.4.8 to reflect changes made in chapters 13 and 14.</li> </ul>
2	<ul style="list-style-type: none"> <li>• Figure 2-1: Transparency in Scrum is updated to include all the key Scrum meetings which facilitate transparency in Scrum projects.</li> <li>• Figure 2-2: Inspection in Scrum is updated to reflect changes made in the names of Scrum processes.</li> <li>• Figure 2-3: Adaptation in Scrum is updated to reflect changes made in the names of Scrum meetings.</li> <li>• In section 2.4, 'servant leadership' is changed to 'supporting leadership.'</li> <li>• Figure 2-5: Goals of a Self-organizing Team is updated to clearly reflect the objectives of self-organization.</li> <li>• Section 2.5.3 is updated to show the use of AI-powered Scrum Project Tool in facilitating collaboration in distributed teams.</li> <li>• Simplified the verbiage for the Three Daily Questions in the <i>Conduct Daily Standup</i> process to be more generic to meeting time of day (section 2.7.1).</li> <li>• Provided more detailed description of the Sprint Planning Meeting (section 2.7.1).</li> <li>• Figure 2-8: Time-Box Durations for Scrum Meetings is updated to depict the time-boxes of the meetings for ease of understanding.</li> </ul>

Chapter	Key Changes Made
3	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Figure 3-1: Scrum Roles—Overview is updated to depict collaboration among Scrum core roles and business stakeholders.</li> <li>• Section 3.2.2 is updated to include ‘Business Stakeholders’ and ‘Supporting Services’ as non-core roles.</li> <li>• Section 3.7 is updated to reflect changes made in chapter 13 Scaling Scrum for Large Projects and chapter 14 Scaling Scrum for the Enterprise.</li> <li>• Figure 3-4: Scrum across the Organization for Projects, Programs, and Portfolios is updated to illustrate how Scrum can be used across the organization for projects, programs, and portfolios.</li> <li>• Summary of Responsibilities (section 3.8) updated to include responsibilities associated with Chief Product Owner, Chief Scrum Master, Program Product Owner, Program Scrum Master, Portfolio Product Owner, and Portfolio Scrum Master roles.</li> <li>• Section 3.9.4 is updated to reflect the change of ‘Servant Leadership’ to ‘Supporting Leadership.’</li> <li>• Section 3.9.5 is updated to include ‘Theory Z.’</li> </ul>
4	<ul style="list-style-type: none"> <li>• Figure 4-3: Business Justification and the Project Lifecycle is updated to accurately summarize the steps in determining business justification.</li> <li>• Section 4.5.2 is updated to include more information on the tool ‘value stream mapping.’</li> <li>• Section 4.5.4 is updated to include more information on the tool ‘story mapping.’</li> <li>• Summary of Responsibilities (section 4.8) updated to include responsibilities associated with Chief Product Owner, Chief Scrum Master, Program Product Owner, Program Scrum Master, Portfolio Product Owner, and Portfolio Scrum Master roles.</li> </ul>
5	<ul style="list-style-type: none"> <li>• Added ‘Definition of Ready’ as a new section 5.4.2.</li> <li>• Improved description of Definition of Done and moved to section 5.4.3.</li> <li>• Improved description of Minimum Done Criteria and moved to section 5.4.4.</li> <li>• Summary of Responsibilities (section 5.6) updated to include responsibilities associated with Chief Product Owner, Chief Scrum Master, Program Product Owner, Program Scrum Master, Portfolio Product Owner, and Portfolio Scrum Master roles.</li> </ul>
6	<ul style="list-style-type: none"> <li>• Minor changes to update terminology to match updates made in other chapters.</li> <li>• Figures 6-4, 6-6, 6-7, and 6-8 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> <li>• Summary of Responsibilities (section 6.7) updated to include responsibilities associated with Chief Product Owner, Chief Scrum Master, Program Product Owner, Program Scrum Master, Portfolio Product Owner, and Portfolio Scrum Master roles.</li> </ul>

Chapter	Key Changes Made
7	<ul style="list-style-type: none"> <li>• Minor changes to update terminology to match updates made in other chapters.</li> <li>• Figure 7-6 is updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> <li>• Summary of Responsibilities (section 7.7) updated to include responsibilities associated with Chief Product Owner, Chief Scrum Master, Program Product Owner, Program Scrum Master, Portfolio Product Owner, and Portfolio Scrum Master roles.</li> </ul>
8	<ul style="list-style-type: none"> <li>• Process 'Identify Scrum Master and Stakeholders' is changed to 'Identify Scrum Master and Business Stakeholder(s).'</li> <li>• Removed following inputs in the Initiate phase: <ul style="list-style-type: none"> <li>○ Program Product Owner</li> <li>○ Program Scrum Master</li> <li>○ Program Stakeholder(s)</li> <li>○ Program Product Backlog</li> </ul> </li> <li>• In 'Identify Scrum Master and Business Stakeholders' process, tool 'Training and Training Cost' is changed to 'Training;' and tool 'Resource Costs' is changed to 'Resource Costing.' Output 'Identified Stakeholders' is changed to 'Identified Business Stakeholder(s).'</li> <li>• In 'Form Scrum Team' process, tool 'Scrum Team Selection' is changed to 'Scrum Team Selection Criteria;' tool 'Training and Training Cost' is changed to 'Training;' and tool 'Resource Costs' is changed to 'Resource Costing.' Tool 'AI-powered Scrum Project Tool' is added. Output 'Back-up Persons' is changed to 'Back-ups.'</li> <li>• Input 'Stakeholders' to 8.4, 8.5, and 8.6 processes is changed to 'Business Stakeholder(s).'</li> <li>• Tool 'AI-powered Scrum Project Tool' is added to 8.4 and 8.5 processes.</li> <li>• Outputs Definition of Ready, High-Level Estimates for Epics, and Dependencies are added to the 'Create Prioritized Product Backlog.'</li> <li>• Figures 8-1 to 8-16 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> <li>• Moved descriptions for Program Product Owner and Program Scrum Master to chapter 3 for consistency.</li> <li>• Minor changes to update terminology and figures to match updates made in other chapters.</li> </ul>

Chapter	Key Changes Made
9	<ul style="list-style-type: none"> <li>• A new tool, “Estimation Methods” is defined to consolidate many of the estimation techniques called out individually in the previous edition (section 9.2.2.1, 9.5.2.3).</li> <li>• The “Create Tasks” process is renamed to Identify Tasks, to clarify that tasks are defined or identified based on the previously Committed User Stories.</li> <li>• ‘Create Sprint Backlog’ process is changed to ‘Update Sprint Backlog’ process.</li> <li>• Inputs, tools, and outputs for all the processes in the Plan and Estimate phase were evaluated and adjusted for correctness.</li> <li>• AI-powered Scrum Project Tool is added as an optional tool in all the processes of this phase.</li> <li>• Definition of Ready is added as a mandatory input to Create User Stories process.</li> <li>• Pre-existing Estimates for User Stories is added as an optional input to Estimate User Stories process.</li> <li>• Sprint Backlog and Scrumboard are added as outputs of Commit User Stories process.</li> <li>• Effort Estimated Task list, output of Estimate Tasks process, is changed to Updated Task list.</li> <li>• Figures 9-1 to 9-20 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> </ul>
10	<ul style="list-style-type: none"> <li>• 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).</li> <li>• Minor changes to update terminology and figures to match updates made in other chapters.</li> <li>• Groom Prioritized Product Backlog is changed to Refine Prioritized Product Backlog.</li> <li>• AI-powered Scrum Project Tool is added as an optional tool in all the processes of this phase.</li> <li>• Sprint Burndown Chart is changed to Sprint Burndown or Burnup Chart across the chapter.</li> <li>• Optional input Stakeholder(s) is changed to Business Stakeholder(s) in the process Refine Prioritized Product Backlog.</li> <li>• Figures 10-1 to 10-10 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> </ul>
11	<ul style="list-style-type: none"> <li>• Minor changes to update terminology and figures to match updates made in other chapters.</li> <li>• Mandatory tool User Story Acceptance/Rejection is added to the process Demonstrate and Validate Sprint.</li> <li>• AI-powered Scrum Project Tool is added as an optional tool in the processes of this phase.</li> <li>• Outputs Accepted Deliverables and Rejected Deliverables are changed to Accepted User Stories and Rejected User Stories respectively in the process Demonstrate and Validate Sprint.</li> <li>• Figures 11-1 to 11-7 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li> </ul>

Chapter	Key Changes Made
12	<ul style="list-style-type: none"><li>• Minor changes to update terminology and figures to match updates made in other chapters.</li><li>• Process Retrospect Project is changed to Retrospect Release.</li><li>• AI-powered Scrum Project Tool is added as an optional tool in the processes of this phase.</li><li>• Tool Retrospect Project Meeting in the process Retrospect Release is changed to Retrospect Release Meeting.</li><li>• Figures 12-1 to 12-7 are updated to reflect changes made to process names, Scrum meetings, artifacts, and concepts in other chapters.</li></ul>
13	<ul style="list-style-type: none"><li>• Scaling Scrum for Large Projects—processes are replaced with additional inputs, tools, and outputs.</li></ul>
14	<ul style="list-style-type: none"><li>• Scaling Scrum for the Enterprise—new processes ‘Create/Update Program or Portfolio Releases’ and ‘Create/Update Program or Portfolio Teams’ are added. Process ‘Create Program or Portfolio Components’ is changed to ‘Create/Update Program or Portfolio Components.’ Process ‘Create and Groom Program or Portfolio Teams’ is changed to ‘Create/Refine Prioritized Program or Portfolio Backlog.’ Process ‘Coordinate Program or Portfolio Components’ is removed.</li></ul>

## The Essential Guide to Successfully Deliver Projects using Scrum

A Guide to the Scrum Body of Knowledge (SBOK® Guide) provides comprehensive guidelines for the successful implementation of Scrum—the most popular Agile product development and project delivery approach. Defined in the SBOK® Guide as a flexible framework, Scrum can be applied to portfolios, programs, or projects of any size or complexity across industries to deliver products, services, or other results.

This Fifth Edition is based on the collective knowledge gained from thousands of projects across diverse organizations and industries. It reflects contributions from a large number of experts in Scrum and project delivery. Feedback from the global Scrum community played a vital role in shaping improvements and additions, making the SBOK® Guide a truly collaborative effort.

Unlike other Scrum references, the SBOK® Guide is available for free on [Scrumstudy.com](https://www.scrumstudy.com), along with free certifications, webinars, videos, and study guides. It is ideal for professionals seeking a foundational understanding of Business Analysis or exploring careers in related fields. The Guide addresses real-life challenges faced by Scrum practitioners and explains how to solve them using modern tools and Artificial Intelligence (AI).

Designed to be accessible and engaging, the SBOK® Guide follows the 80-20 rule—80% of key concepts can be learned by reading just 20% of the content, with the remainder available for deeper reference. It is more readable than most Scrum books, which are often either too simplistic or overly detailed.

The SBOK® Guide serves as a reference for both experienced practitioners and those with no prior knowledge of Scrum or project delivery methods. Organized for easy navigation, the SBOK® Guide aims to inform, support, and inspire all readers through its rich, collaborative content.

