SCRUMstudy
Targeting success

A Guide to the

SCRUM BODY OF KNOWLEDGE

(SBOK® Guide)

6. CHANGE

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

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

6. CHANGE

6.1 Introduction

Every project, regardless of the method or framework used to deliver it, 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.

Change, 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 business 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 members in several teams.

This chapter is divided into the following sections:

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

6.2 Overview—This section defines the concept of change, specifically within the context of Scrum processes. It also addresses how Change Requests are handled in Scrum processes.

6.3 Change in Scrum—This section details the importance of effectively managing change in a Scrum project. It also addresses how flexibility and stability can be achieved through appropriate handling of the Change Requests that arise throughout a project.

6.4 Integrating Change—This section details how Change Requests are assessed and approved (or rejected) when applying the Scrum framework.

6.5 Change to Programs and Portfolios—This section describes the impact of changes to programs and portfolios.

6.6 Summary of Responsibilities—This section defines the change management responsibilities of project team members.

6.7 Scrum vs. Traditional Project Management—This section discusses the benefits of managing change using Scrum methods over the methods used in traditional project management models.

6.1.1 Roles Guide

1. Product Owner—The responsibility of initiating change in a project lies primarily with the Product Owner; therefore, this entire chapter is applicable to this role.
2. Scrum Master—The Scrum Master should also be familiar with this entire chapter with primary focus on sections 6.3, 6.4, 6.5, and 6.7.
3. Scrum Team—The Scrum Team should focus on sections 6.3, 6.4.2, 6.5, and 6.7.

6.2 Overview

Change is inevitable in all projects. In today's hypercompetitive world where technology, market conditions, and business patterns are continuously shifting, change is the only constant.

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

Scrum development projects welcome change by using small development cycles that incorporate customer feedback on the project's deliverables after each Sprint. This enables the customer to regularly interact with the Scrum Team members, view product increments as they are ready, and change requirements earlier on in the development cycle. Also, the portfolio or program management teams can respond to Change Requests pertaining to Scrum projects applicable at their level.

Scrum embodies a key principle from the Agile Manifesto (Fowler and Highsmith, 2001): "Responding to change over following a plan." Scrum is practiced on the basis of embracing change and turning it into a competitive advantage. Therefore, it is more important to be flexible than to follow a strict, predefined plan. This means it is essential to approach project management in an adaptive manner that enables change throughout rapid product development or service development cycles.

Being adaptive to change is a key advantage of the Scrum framework. Although Scrum works well for all projects in all industries, it can be very effective when the product or other project requirements are not fully understood or cannot be well defined up front, when the product's market is volatile, and/or when the focus is on making the team flexible enough to incorporate changing requirements. Scrum is especially useful for complex projects with a lot of uncertainty. Long-term planning and forecasting is typically ineffective for such projects, and they involve high quantities of risk. Scrum guides the team through transparency, inspection, and adaptation to the most valuable business outcomes.

6.2.1 Unapproved and Approved Change Requests

Requests for changes are usually submitted as Change Requests. Change Requests remain unapproved until they get formally approved. The Scrum Guidance Body usually defines a process for approving and managing changes throughout the organization. In the absence of a formal process, it is recommended that minor changes that do not have significant impact on the project be directly approved by the Product Owner. The tolerance for such minor changes could be defined at an organizational level or by the sponsor for a particular project. In most projects, 90% of Change Requests could be classified as minor changes that should be approved by the Product Owner. So, the Product Owner plays a key role in managing changes in a Scrum project. Changes that are beyond the approval level of the Product Owner may need approval from relevant business stakeholders working with the Product Owner.

At times, if a requested change could have a substantial impact on the project or organization, approval from senior management (e.g., Executive Sponsor, Chief Product Owner, Program Product Owner, and/or Portfolio Product Owner) may be required. Change Requests for the project are discussed and approved during the *Develop Epic(s)*, *Create Prioritized Product Backlog,* and *Refine Prioritized Product Backlog* processes. *Approved Change Requests* are then prioritized along with other product requirements and their respective User Stories and then incorporated into the Prioritized Product Backlog.

Figure 6-1 illustrates a sample change approval process, outlining steps for evaluating, reviewing, and approving change requests within Scrum projects.

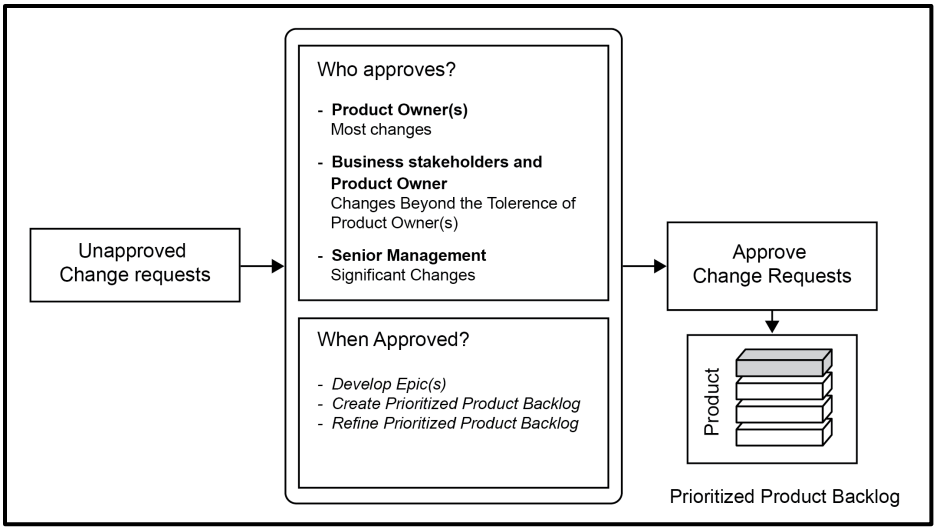

Figure 6-1: Sample Change Approval Process

Figure 6-2 shows how approved changes are updated in the Prioritized Product Backlog, ensuring real-time backlog refinement and transparency.

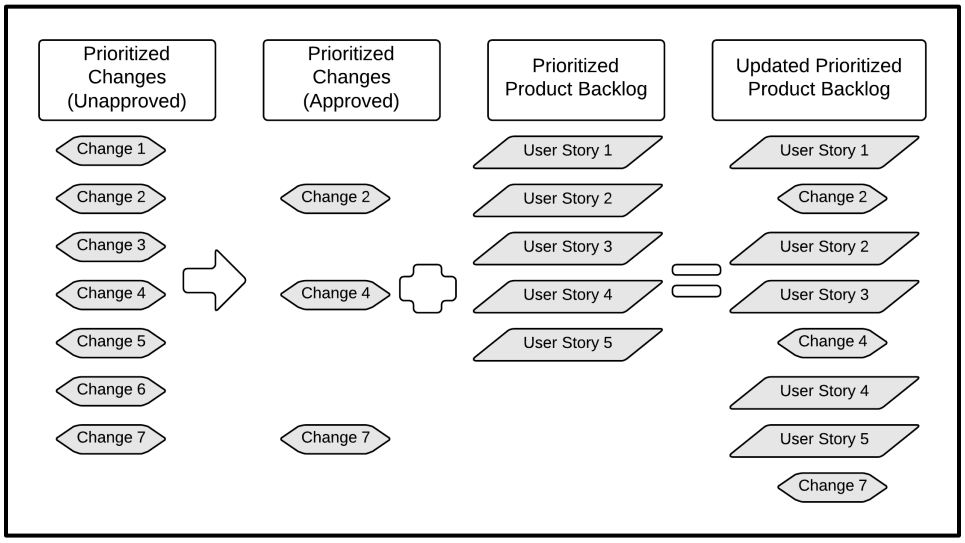

Figure 6-2: Updating Prioritized Product Backlog with Approved Changes

6.3 Understanding Change in Scrum

6.3.1 Balancing Flexibility and Stability

Scrum helps organizations become more flexible and open to change. However, it is important to understand that although the Scrum framework emphasizes flexibility, it is also important to maintain stability throughout the change process. In the same way that extreme rigidity is ineffective, extreme flexibility is also unproductive. The key is to find the right balance between flexibility and stability because stability is needed in order to get work done. Therefore, Scrum uses iterative delivery and its other characteristics and principles to achieve this balance. Scrum maintains flexibility in that Change Requests can be created and approved at any time during the project; however, they get prioritized when the Prioritized Product Backlog is created or updated. At the same time, Scrum ensures that stability is maintained by keeping the Sprint Backlog fixed and by not allowing interference with the Scrum Team during a Sprint. In Scrum, all requirements related to an ongoing Sprint are frozen during the Sprint. No change is introduced until the Sprint ends, unless a change is deemed to be significant enough to stop the Sprint. In the case of an urgent change, the Sprint is terminated, and the team meets to plan a new Sprint. This is how Scrum accepts changes without creating the problem of changing release dates.

6.3.2 Incorporating Flexibility

Because of its iterative nature and the empirical process control concepts of transparency, inspection, and adaptation, Scrum implementation must incorporate flexibility. Scrum provides an adaptive mechanism for project management in which a change in requirements can be accommodated without significantly impacting overall project progress. It therefore becomes possible (and necessary) to adapt to emerging business realities as part of the development cycle. Flexibility in Scrum is achieved through five key characteristics—iterative product development, Time-boxing, cross-functional teams, customer value-based prioritization, and continuous integration (see Figure 6-3).

Figure 6-3 highlights Scrum characteristics for achieving flexibility, including iterative delivery, continuous feedback, and adaptive planning.

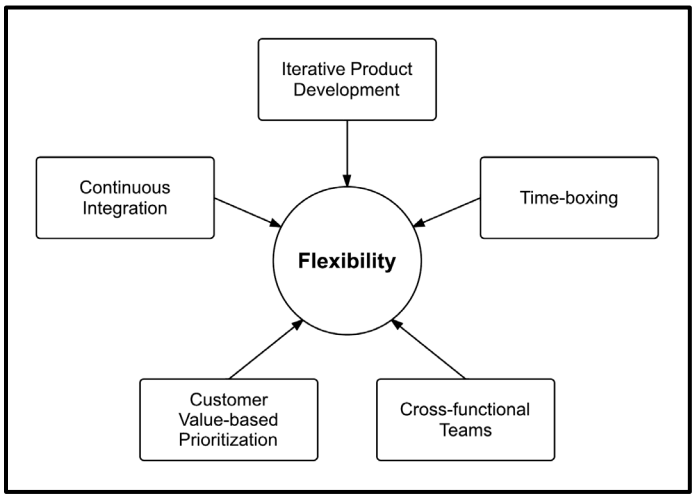

Figure 6-3: Scrum Characteristics for Achieving Flexibility

6.3.2.1 Flexibility through Iterative Product Development

Scrum follows an iterative and incremental approach to product and service development, making it possible to incorporate change at any step in the development process. As the product is developed, a Change Request for the project can come from multiple sources as follows:

1. **Business stakeholders**

   Business stakeholders—particularly sponsors, customers, and users—may submit Change Requests at any time throughout the project. Change Requests could be due to changes in market conditions, organizational direction, legal or regulatory issues, or various other reasons. Moreover, business stakeholders may submit Change Requests as they are reviewing the deliverables during the *Demonstrate and Validate Sprint*, *Retrospect Sprint*, or *Retrospect Release* processes. All Change Requests get added to the Project Prioritized Product Backlog (also referred to as Prioritized Product Backlog or Product Backlog) once approved.

   Figure 6-4 shows the motivation of business stakeholders for requesting changes, including evolving market demands, customer feedback, and compliance needs.

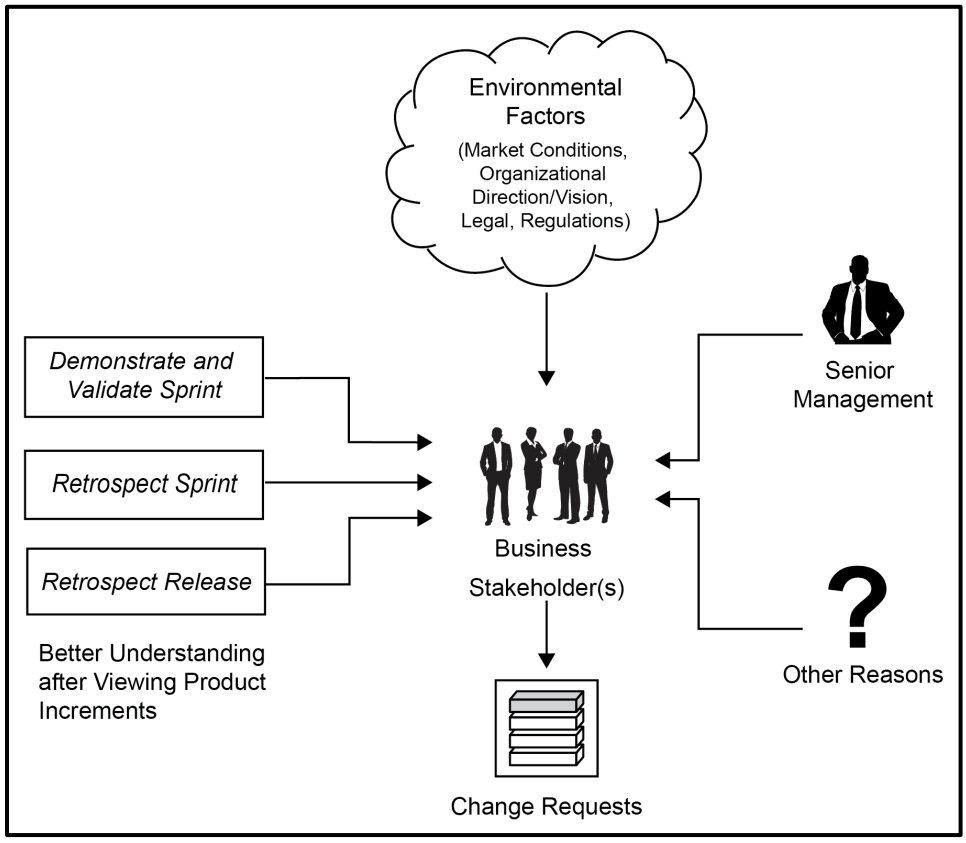

Figure 6-4: Motivation of Business Stakeholders for Requesting Changes

2. Scrum Core Team

The Scrum Core Team (i.e., the Product Owner, Scrum Master, and Scrum Team) are involved in creating the product deliverables. Ongoing interaction between the Scrum Core Team members in a Scrum Team and others, such as other Scrum Teams in the project, and internal and external project business stakeholders, may motivate Scrum Core Team members to suggest changes or improvements to the product, service, or some other part of the project. Usually such changes—like any others—are captured in Change Requests, and the Product Owner makes a final decision about which suggested changes from the Scrum Team and Scrum Master should be considered as formal Change Requests.

There may at times be challenges with creating certain deliverables, which may result in Change Requests. For example, the team may decide on a new feature to be added or modified to improve product performance. In most Scrum projects, recommendations for changes from the Scrum Core Team happen as Scrum Teams work to *Create Deliverables*, or when they participate in meetings associated with the *Conduct Daily Standup* or *Retrospect Sprint* processes.

Figure 6-5 displays the motivation of the Scrum Core Team for requesting changes, focused on technical improvements, team learning, and value delivery optimization.

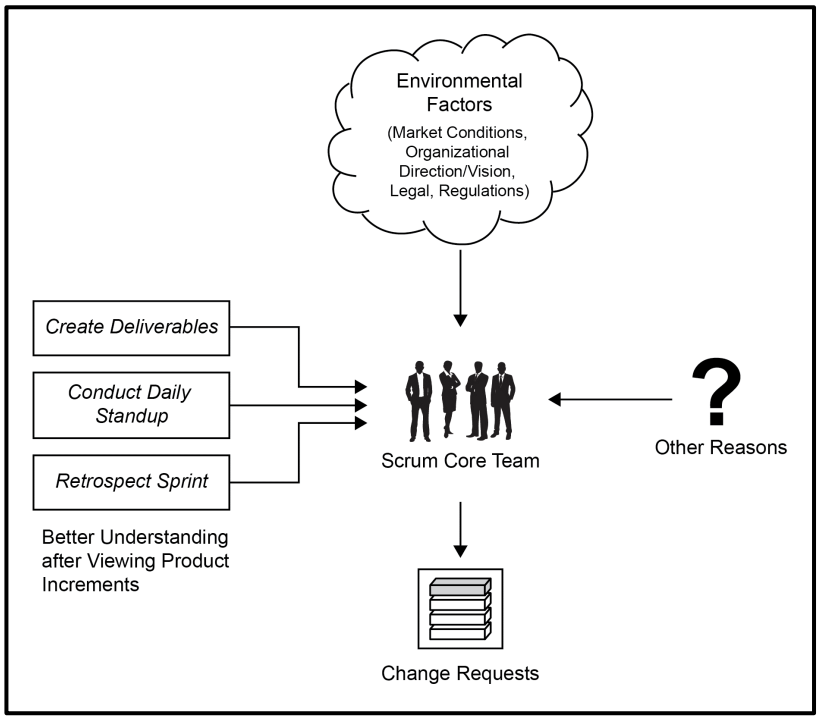

Figure 6-5: Motivation of Scrum Core Team for Requesting Changes

3. Senior Management

Senior management—including portfolio and program management—can recommend changes that affect the project. This can be because of changes in the strategic direction of the organization, competitive landscape, funding-related issues, and so forth.

Note that such changes get added to the Prioritized Product Backlog and need to go through the usual change management process. If any of these changes are urgent, any impacted Sprint may need to be terminated.

4. **Scrum Guidance Body**

   The Scrum Guidance Body may submit Change Requests that affect all projects due to any of the following examples:

   - Change in government regulations (e.g., privacy, security standards, or new legislation)
   - Corporate directives for quality, security, or other organizational initiatives that need to be implemented across the organization
   - Benchmarks or best practices to meet a certain standard
   - Lessons learned from previous projects, which could be implemented by other Scrum Teams

   The hallmark of Scrum is that it is change tolerant and adaptive. Scrum does not promote determining and firmly setting plans way in advance because it operates on the premise that project development is extremely prone to change and risk. The result is a high degree of flexibility and tolerance for change. The project is planned, executed, and managed incrementally, so it is typically easy to incorporate changes throughout.

6.3.2.2 Flexibility through 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. Examples of Time-boxing include limiting the Daily Standup Meetings to 15 minutes and setting Sprint durations to be one to four weeks. Time-boxed Sprints contribute toward meeting deadlines and achieving elevated levels of productivity. Errors or problems can be identified early and rectified quickly, enabling flexibility in Scrum projects. Sprints promote order and consistency in a volatile work environment. They provide a platform to gauge results and obtain feedback in a short span of time. Sprints also allow for frequent assessment of progress and the methods used to manage the project, including effective change management.

Furthermore, by incorporating Time-boxing in Sprints, the team frequently revisits the process of estimating the work to be done, so the projection of time and effort required becomes more accurate with each subsequent Sprint as the project progresses. These iterative cycles also motivate team members to achieve the projected targets and incremental goals toward reaching the larger project objectives.

6.3.2.3 Flexibility through Cross-functional and Self-organized Teams

Self-organization ensures that Scrum Team members have the flexibility to determine all the tasks they will work on in a Sprint and how they will complete the work. It also keeps teams self-motivated to complete their self-assigned tasks, removes bottlenecks, and encourages the sharing of knowledge with other team members.

The cross-functional and self-organized structures of the Scrum Team allow team members to be extremely focused on the desired Sprint results. The team has a defined set of objectives during each Sprint and the flexibility to account for a change in objectives prior to beginning the next Sprint.

The use of cross-functional teams also ensures that all of the skills and knowledge required to carry out the work of the project exists within the team itself. This provides an efficient working model that result in the creation of deliverables that are potentially shippable and ready for demonstration to the Product Owner and/or other business stakeholders.

Self-organization ensures that Scrum Team members determine on their own *how* to do the work of the project without a senior manager micromanaging their tasks.

Having cross-functional and self-organized teams allows the group to adapt and effectively manage the ongoing work and any minor issues or changes without having to obtain support or expertise from members outside the team, while in the process of creating deliverables.

6.3.2.4 Flexibility through Customer Value-based Prioritization

The prioritization of requirements and work in a Scrum project is always determined based on the value provided to the customer. First, at the start of a project, the initial requirements are prioritized based on the value each requirement will provide—this is documented in the Prioritized Product Backlog. Then, when a request is made for a new requirement or a change to an existing one, it is evaluated during the *Refine Prioritized Product Backlog* process. If the change is deemed to provide more value than other existing requirements, it will be added and prioritized accordingly in the updated Prioritized Product Backlog. So, the Prioritized Product Backlog provides scope for incorporating changes and adding new requirements when necessary.

It is important to note that new requirements and changes added to the Prioritized Product Backlog may lower the priority of other existing User Stories in the Backlog: so, such lower prioritized User Stories may be implemented later depending on their new prioritization. Because customers are very closely involved with the prioritization of requirements and their corresponding User Stories in the Prioritized Product Backlog, this practice ensures that the requirements that customers deem as “high value” get completed sooner and that the project starts delivering significant value much earlier on.

6.3.2.5 Flexibility through Continuous Integration

Using continuous integration techniques, Scrum Team members can incorporate new and modified features into the deliverables whenever possible. This mitigates the risk of multiple team members making changes to redundant components (e.g., obsolete code in software products, old designs for manufacturing parts). This ensures that only the latest feature or version is being worked on and avoids compatibility issues.

6.4 Integrating Change

Depending on the industry and type of project, the priority of features and requirements for a project may remain fixed for significant durations of time, or they may change frequently. If project requirements are generally stable, there are typically only minor changes made to the Prioritized Product Backlog throughout development, and Scrum Teams can work sequentially completing requirements that will provide maximum customer value as prioritized in the Prioritized Product Backlog. The length of the Sprint is usually longer, 4 to 6 weeks, in such stable environments. If project requirements change throughout the project, for example, due to changed business requirements, the same method continues to be effective. Before beginning a Sprint—during the *Create Prioritized Product Backlog* or *Refine Prioritized Product Backlog* processes—the highest priority requirements in the Prioritized Product Backlog are typically selected to be completed in that Sprint. Because changes have been accounted for in the Prioritized Product Backlog, the team only needs to determine how many tasks they can accomplish in the Sprint based on time and resources provided. Change management is executed in the ongoing processes of prioritizing and adding tasks to the Prioritized Product Backlog.

6.4.1 Changes to a Sprint

If there is a Change Request that may have significant impact on a Sprint in progress, the Product Owner, after consultation with relevant business stakeholders, decides whether the change can wait until the next Sprint or represents an urgent situation which may require ending the current Sprint and starting a new one. The Scrum framework clearly specifies that the scope of a Sprint cannot be changed once the Sprint begins. If the required change is so important that the results of the Sprint would be worthless without it, then the Sprint should be terminated. If not, then the change is incorporated into a later Sprint (as shown in Figure 6-6).

Figure 6-6 shows how a Product Owner manages change requests during a sprint in an Agile framework.

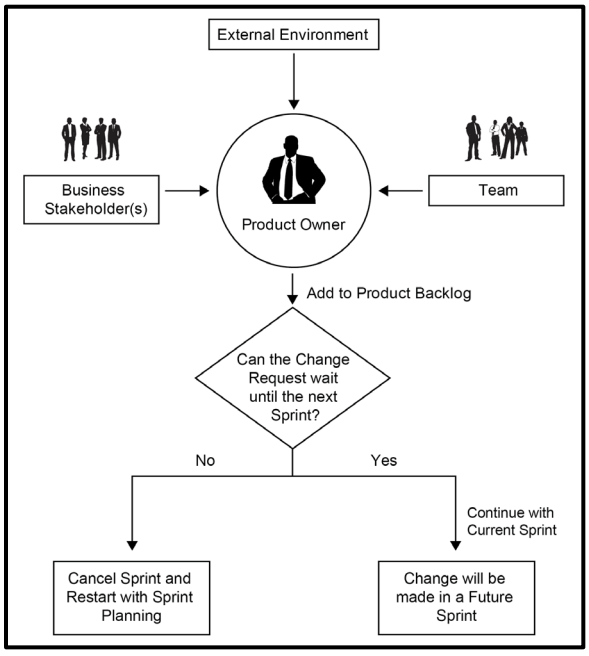

Figure 6-6: Incorporating a Change during a Sprint

There is only one exception to this rule about not changing the scope of a Sprint once a Sprint begins. If the Scrum Team determines it has heavily overestimated the effort during the Sprint and has spare capacity to implement additional User Stories, the team can ask the Product Owner which additional User Stories can be included in the current Sprint. By locking down the scope of every Sprint, the team is able to efficiently optimize and manage their work and effort. An additional benefit is that the team does not have to worry about managing changes once they start working on a Sprint. This is a big advantage of the Scrum framework as compared with traditional project management.

In traditional project management, changes can be requested and approved anytime during the project's lifecycle. This often creates confusion for project team members, decreases team motivation due to discontinuity, and results in a lack of focus and the team feeling that "nothing ever gets done." On the other hand, in Scrum projects, changes are not allowed once a Sprint starts. This ensures that in every Sprint, the team completes deliverables and tasks are Done. Furthermore, the business recognizes tangible benefits from potentially shippable deliverables at the end of each Sprint.

Moreover, since the Product Owner and business stakeholders are aware that changes are not allowed once a Sprint begins and a Sprint lasts between 1 and 4 weeks, they define and prioritize requirements during the appropriate processes of *Create Epic(s)*, *Create Prioritized Product Backlog,* and *Refine Prioritized Product Backlog.*

6.4.1.1 Impact of Expected Change on the Length of Sprint

Because changes are not allowed during a Sprint, the likelihood of any changes arising may have an impact on the decision related to the Length of Sprint when it is determined during the *Conduct Release Planning* process.

Figure 6-7 demonstrates an inverse relationship between sprint length and the probability of change during the sprint.

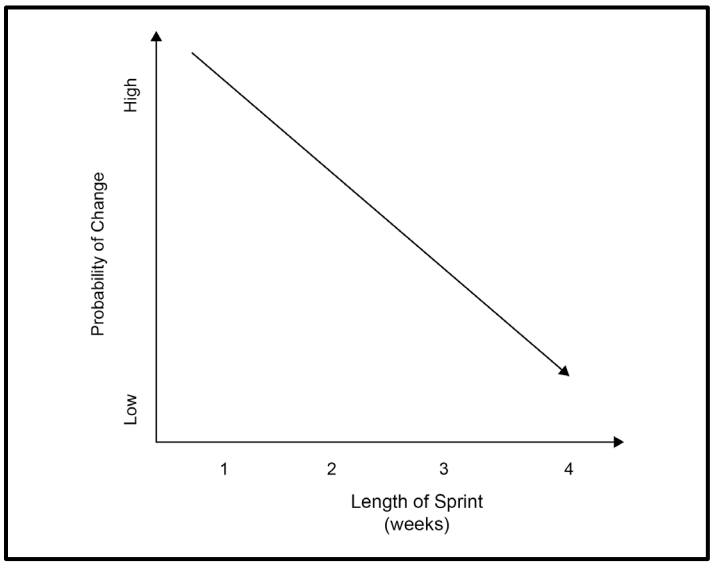

Figure 6-7: Impact of the Probability of Change on the Length of Sprint

A Sprint is Time-boxed with duration of one to four weeks. Most Scrum projects typically have Sprints Time-boxed at duration of two or three weeks. If project requirements are not very well defined, or if significant changes are expected in the immediate future, the Length of Sprint is typically set at one to three weeks. This provides enough stability to the Scrum Team members to work on shorter Sprints, and in turn, deliver results quickly (which are then evaluated by the Product Owner and business stakeholders at the end of each Sprint). This also provides enough flexibility for the team to clarify requirements and make changes to the Prioritized Product Backlog at the end of each Sprint.

However, if project requirements are stable and major changes are not expected in the near future, the Length of a Sprint may in some cases be set to extend up to six weeks. This provides more stability to the Scrum Team members to work on the Prioritized Product Backlog requirements for lengthy periods of time without having to prematurely go through the *Create User Stories*, *Estimate User Stories, Commit User Stories*, *Identify Tasks*, *Estimate Task*, and other related processes that are conducted for every Sprint. It is important to note that the likelihood of any change arising is not the only factor that should be used to determine the Length of Sprint. Other factors that also need to be considered include:

- Actual time to get the work done (if the project or corporate environment needs a specific time to get tasks done, this could determine the Length of Sprint)
- Planned date for a release (the Length of Sprint should take into consideration the release dates for the overall product or service)
- Any other factor as determined by the Product Owner or Scrum Master, that needs to be considered while determining the Length of Sprint

Changing the Length of Sprint should not be decided lightly or periodically. For example, it is not advisable to have the Sprint length as three weeks for the current Sprint, two weeks for the next Sprint, four weeks for the third Sprint, and so on. Length of Sprint should preferably be consistent. One of the greatest impacts of changing the Length of Sprint is that it causes a reset on all tracking at the project level. Previous velocities may become useless for forecasting and planning of future Sprints. Without an accurate velocity (which is a primary metric in any Scrum project), the Scrum Team cannot be measured for effectiveness or adequately choose the number of User Stories to take on when planning for a Sprint. So, once the Length of Sprint is decided, it should preferably be kept constant over the duration of the project or through multiple Sprint cycles.

6.4.1.2 Managing Changes through Prioritized Product Backlog Refining

A typical Prioritized Product Backlog will contain all User Stories, their time estimates (including any revised estimates), and the status of higher priority requirements. Any new or revised User Stories resulting from changes to business requirements, customer requests, external market conditions, and/or lessons learned from previous Sprints are also incorporated. One of the Product Owner's key responsibilities is refining the Prioritized Product Backlog to ensure the prioritized requirements in the Prioritized Product Backlog to be included in the next two to three Sprints are refined into suitable User Stories. It is recommended that the Product Owner should spend a significant amount of the time in each Sprint for Prioritized Product Backlog refining. The Product Owner is responsible for adding and revising Prioritized Product Backlog Items in response to any changes and is responsible for providing more detailed User Stories that will be used for the next Sprint.

Refining helps ensure that refining of requirements and their User Stories is done well in advance of the Sprint Planning Meeting so that the team has a well-analyzed and clearly defined set of stories that can be easily broken down into tasks and subsequently estimated. Based on lessons learned from the current Sprint, there may be changes to requirements, or there may be reprioritization that can be easily incorporated into subsequent Sprints. Refining supports and enhances the flexibility of the Scrum model by incorporating the latest business and technical insights into future Sprints.

A Product Backlog Review Meeting (also referred to as a Prioritized Product Backlog Refining Session) is a formal meeting during the *Refine Prioritized Product Backlog* process, which helps the Scrum Team review and gain consensus about the Prioritized Product Backlog. However, other than the Prioritized Product Backlog Review Meeting, Prioritized Product Backlog refining should happen throughout the project and can include situations in which the Product Owner writes new User Stories or reprioritizes User Stories in the existing Prioritized Product Backlog, Scrum Team members or business stakeholders give their suggestions about new User Stories to the Product Owner, and so forth. It is important to note that any item in the Prioritized Product Backlog is always open for re-estimation until the Sprint Backlog is finalized in the *Update Sprint Backlog* process. After that, changes can continue to be made until immediately prior to the Sprint Planning Meeting, if required.

6.4.1.2.1 Effective Product Backlog Review Meeting (or Prioritized Product Backlog Refining Session)

The Product Owner takes the lead in a Product Backlog Review Meeting which is conducted during the *Refine Prioritized Product Backlog* process. It is important that the Product Owner sets the objectives and ideally develop an agenda before the Product Backlog Review Meeting begins. Without the objectives, the session will be unstructured and may prove unproductive. It is also important to limit the number of business stakeholders participating in the meeting. Having too many participants tends to decrease the overall efficiency of the meeting. The Product Owner should invite only those business stakeholders whose feedback is required for the refining session. All Scrum Team members should be included because their input is valuable to the work being done and any issues encountered. If the refining session results in any reprioritization of or change in the Prioritized Product Backlog, it is important that the team agrees with those changes. An effective refining session should result in clearly defined Prioritized Product Backlog Items (PBIs) so that the Scrum Team clearly understands what the customer's requirements are. This also helps the team become familiar with all User Stories in case one or more of them needs to be included in a Sprint at short notice. Acceptance and Done Criteria may also be discussed during refining sessions. Scrum does not Time-box refining exercises. Prioritized Product Backlog refining is a continuous activity for the Product Owner.

6.4.1.3 Managing Changes During Demonstrate and Validate Sprint

Although the Product Owner has the final say on Prioritized Product Backlog Items and whether to accept or reject any User Stories (corresponding to Approved Change Requests) presented during the *Demonstrate and Validate Sprint* process, it is the Scrum Master's responsibility to ensure that the requirements and Acceptance Criteria are not altered during the Sprint Review Meeting for the User Stories completed in the current Sprint. This prevents the rejection of completed User Stories based on the fact that they do not meet newly changed requirements. If any requirements need to be changed, any corresponding PBI needs to be revised to accommodate the modified requirements in a future Sprint.

6.5 Change in Programs and Portfolios

Any change that arises in either programs or portfolios may have a cascading effect on all dependent projects and their corresponding Sprints. Therefore, it is advisable to minimize changes at these higher levels. If a change is required and all business stakeholders agree to make the change at these levels, the following should be kept in mind.

6.5.1 In Programs

1. It is not recommended to make changes in between two Program Backlog Meetings.
2. If the change is minor, the Program Product Owner should secure approval from the relevant business stakeholders (e.g., sponsor, customer, and end user) and the Portfolio Product Owner and then add the requirements to the Prioritized Program Backlog. Product Owners for the project will consider those requirements for inclusion in future Sprints.
3. If the change is major, the program efforts along with associated projects and Sprints need to stop, and a Prioritized Product Backlog Meeting should be conducted to determine next steps.
4. Prioritized Program Backlog Meetings (also referred to as Program Backlog Meetings) should preferably be conducted at two- to six-month intervals. The frequency and impact of changes to a program determine the time duration between two Program Backlog Meetings. If there are several changes expected to arise in the program, it is preferable to conduct Program Backlog Meetings at more regular intervals (e.g., two to three months); but if there are fewer changes expected and if requirements are stable, the duration between two Program Backlog Meetings could be increased (e.g., five to six months).

6.5.2 In Portfolios

1. It is not recommended to make changes in between two Portfolio Backlog Meetings.
2. If the change is minor, the Portfolio Product Owner should secure approval from the relevant business stakeholders (e.g., sponsor, customer, and end user) and then add the requirements to the Prioritized Portfolio Backlog. Product Owners of the program and project will consider those requirements for inclusion in future Sprints.
3. If the change is major, the portfolio efforts along with associated programs, projects, and Sprints need to stop, and a Portfolio Backlog Meeting should be conducted to determine next steps.
4. Prioritized Portfolio Backlog Meetings (also referred to as Portfolio Backlog Meetings) should be conducted at four- to twelve-month intervals. The frequency and impact of changes to a portfolio determine the time duration between two Portfolio Backlog Meetings. If there are several changes expected in the portfolio, it is preferable to conduct Portfolio Backlog Meetings at more regular intervals (e.g., four to six months); but if there are fewer changes expected and if requirements are stable, the duration between two Portfolio Backlog Meetings could be increased (e.g., nine to twelve months).

Figure 6-8 depicts how changes are incorporated in programs and portfolios. It outlines a structured approach to evaluating, approving, and integrating changes to ensure alignment with organizational goals.

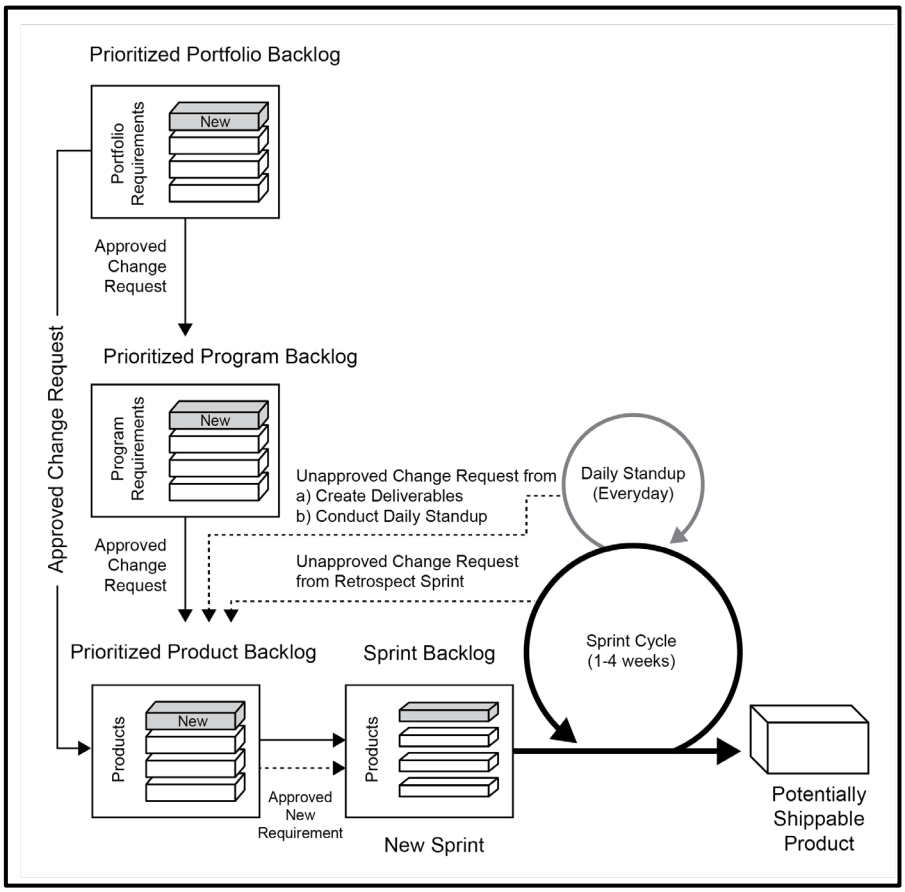

Figure 6-8: Incorporating Changes in Programs and Portfolios

6.6 Summary of Responsibilities

Role	Responsibilities
Scrum Team	• Suggests improvements or changes during the *Create Deliverables* and *Conduct Daily Standup* processes
Product Owner/ Chief Product Owner	• Provides requests for changes in a project • Assesses the impact of requests for change raised for the portfolio, program, or project • Prioritizes User Stories in the project's Prioritized Product Backlog • Assesses the impact of problems on project objectives identified by the Scrum Team • Provides clear communication to business stakeholders on reprioritized Product Backlog Items
Scrum Master/ Chief Scrum Master	• Facilitates identification, assessment, and escalation of problems and Change Requests by the Scrum Team
Program Product Owner	• Provides request for change for programs • Approves products that are amended, removed, or added according to program requirements
Program Scrum Master	• Facilitates identification, assessment, and management of Change Requests for programs
Portfolio Product Owner	• Provides Change Requests for portfolios • Approves products that are amended, removed, or added according to portfolio requirements
Portfolio Scrum Master	• Facilitates identification, assessment, and management of Change Requests for portfolios
Business Stakeholder(s)	• Provides request for changes • Involved with approving and prioritizing Change Requests
Scrum Guidance Body	• Provides overall guidance for the change management procedures to be followed throughout the project

Table 6-1: Summary of Responsibilities Relevant to Change

6.7 Scrum vs. Traditional Project Management

Change management in traditionally managed projects is closely related to Configuration Management. All changes are considered based on their magnitude of variation from a baseline value. The Project Manager is given thresholds within which he or she can manage the day-to-day activities and decisions of the project. When a Change Request exceeds the defined tolerances, the Project Manager must escalate the proposed change to higher levels of management and await their decision before implementing the change. The Project Manager first logs the request for change in an Issue Log or Change Log and then escalates the change to higher authorities. These might include the sponsor of the project, as well as other relevant business stakeholders and decision makers. At some point, an impact assessment will be conducted. Based on the estimated impact of the change, a decision will be made regarding whether the change should be implemented or not. The Project Manager may also propose viable solutions to any problems posed by the change. If the higher authorities decide to proceed with making the change, the Project Manager is responsible for ensuring that the change is implemented correctly.

Change in Scrum works differently as compared with Traditional Project Management. The Scrum framework is highly tuned toward managing changes effectively and efficiently. Whenever the Product Owner or the Scrum Team recognizes a problem or defect or identifies a Prioritized Product Backlog Item that needs to be amended, replaced, or added, the change is made to the Prioritized Product Backlog. Similarly, senior management, the Product Owner, or business stakeholder(s) can add Change Requests to the Prioritized Product Backlog. The Product Owner and business stakeholder(s) approve Change Requests and reprioritize the Prioritized Product Backlog accordingly. Whenever there is a problem or new requirement that needs to be addressed immediately and mandates a change affecting the current Sprint, the Product Owner terminates the Sprint, with approval from relevant business stakeholders. Once terminated, the Sprint will be re-planned and restarted to incorporate the new requirements.

However, if the problem or change is not major and does not warrant a change within the current Sprint, the change will be added to the Prioritized Product Backlog and incorporated into the planning for a subsequent Sprint. This gives business stakeholders the ability to respond to changes in the external environment, while still maintaining a certain degree of control over the ongoing activities within the project. Also, at the end of each Sprint, completed deliverables are demonstrated by the Scrum Team. These deliverables are potentially shippable and can be reviewed by the Product Owner and other business stakeholders.

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, 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.

SCRUMstudy™ is a brand of VMEdu, Inc.