SCRUMstudy
Targeting success

A Guide to the

SCRUM BODY OF KNOWLEDGE

(SBOK® Guide)

10. IMPLEMENT

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

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

10. IMPLEMENT

The Implement phase is related to the execution of the tasks and activities to create a project's product. These activities include creating various deliverables, conducting Daily Standup Meetings, and refining (i.e., reviewing, fine-tuning, and regularly updating) the Product Backlog at regular intervals.

Implement, 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.

To facilitate the best application of the Scrum framework, this chapter identifies inputs, tools, and outputs for each process as either "mandatory" or "optional." Inputs, tools, and outputs denoted by asterisks (*) are mandatory, or considered critical for project success, whereas those with no asterisks are optional.

It is recommended that the Scrum Team and those individuals being introduced to the Scrum framework and processes focus primarily on the mandatory inputs, tools, and outputs; while Product Owners, Scrum Masters, and other more experienced Scrum practitioners strive to attain a more thorough knowledge of the information in this entire chapter.

This chapter is written from the perspective of one Scrum Team working on one Sprint to produce potentially shippable deliverables, which could be part of a larger project, program, or portfolio. Additional information pertaining to Scaling Scrum for Large Projects is available in chapter 13. Additional information pertaining to Scaling Scrum for the Enterprise can be found in chapter 14.

Implement is the second of the three phases that are done repetitively in every Sprint. This phase begins after Sprint planning is complete. It is the core of every Scrum project where the bulk of the work is done.

The Scrum Team, facilitated by the Scrum Master, creates the deliverables that are associated with the committed User Stories by working on and completing the tasks the team identified in the previous phase.

While the Scrum Team is creating the deliverables of the Sprint, the Product Owner updates and refines the Prioritized Product Backlog to keep it up to date with any changes in requirements and/or priorities and to ensure that the set of User Stories the Product Owner would like the team to commit to in the next Sprint will be ready for commitment.

It is also important to realize that although all phases and processes are defined uniquely in the SBOK® Guide, they are not necessarily performed sequentially or separately. At times, it may be more appropriate to overlap some phases and/or processes, depending on the specific requirements of each project.

Figure 10-1 provides an overview of the "implement" phase in scrum. Summarizes key processes like creating deliverables, conducting daily standups, and refining the backlog, along with associated roles and expected outputs.

10.1 Create Deliverables—In this process, the Scrum Team creates the Sprint deliverables by working on the tasks in the Sprint Backlog. This is the process where the Scrum Team and the Scrum Master spend most of their time. The team is supported by the Scrum Master, who facilitates meetings for the team, addresses impediments the team faces, and does whatever he/she can do to allow the Scrum Team members to focus on the creation of the Sprint Deliverables.

The Scrum Team uses a Scrumboard to track its progress during the Sprint. The Scrum Team uses the information about its progress to get a good indication of its ability to deliver according to its commitment and, if necessary, to take action to secure the most valuable outcome of the Sprint that is possible under the given circumstances.

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

10.3 Refine Prioritized Product Backlog—In this process, the Product Owner continuously updates and maintains the Prioritized Product Backlog. A Prioritized Product Backlog Review Meeting may be held, during which any changes or updates to the Product Backlog are discussed and incorporated into the Prioritized Product Backlog as appropriate.

In order to keep the Prioritized Product Backlog up to date with any change in requirements and/or priorities, the Product Owner continually works with the customer and other business stakeholders to capture and understand any changes in their needs.

To ensure that the set of User Stories the Product Owner would like the team to commit to in the next Sprint will be ready for commitment, the Product Owner refines existing Epics and User Stories in the Prioritized Product Backlog, and ensures that the User Stories satisfy the Definition of Ready.

As part of refining the Prioritized Product Backlog, the Product Owner also works with the Scrum Team to get feedback and questions related to the updates in the Prioritized Product Backlog, potentially including estimates.

If changes in requirements and/or the overall progress of the Scrum Team require changes to the Release Schedule and/or the business justification, the Product Owner will also make these changes during this process.

This is the process where the Product Owner will spend most of his/her time.

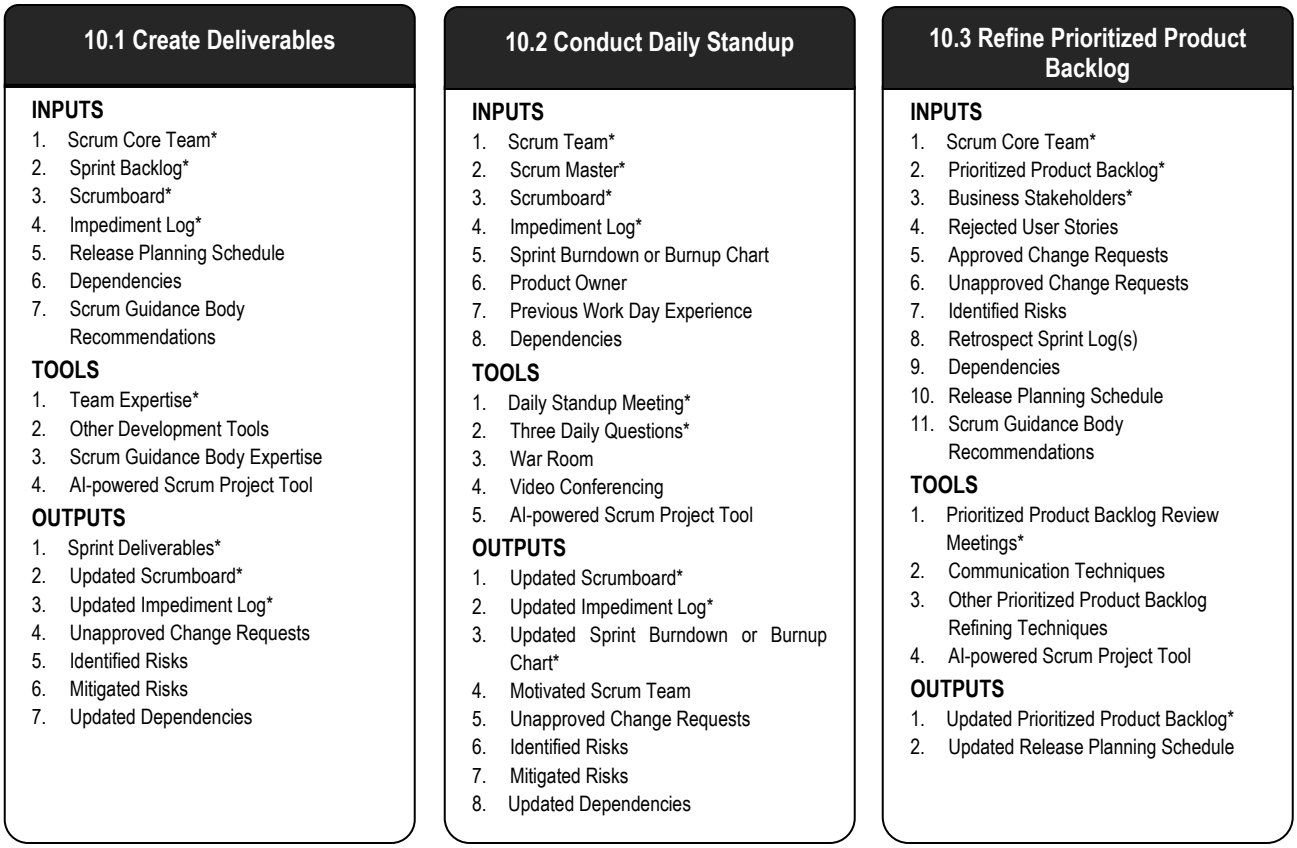

Figure 10-1: Implement Overview

Note: Asterisks (*) denote a "mandatory" input, tool, or output for the corresponding process.

Figure 10-2 below shows the mandatory inputs, tools, and outputs for processes in the Implement phase.

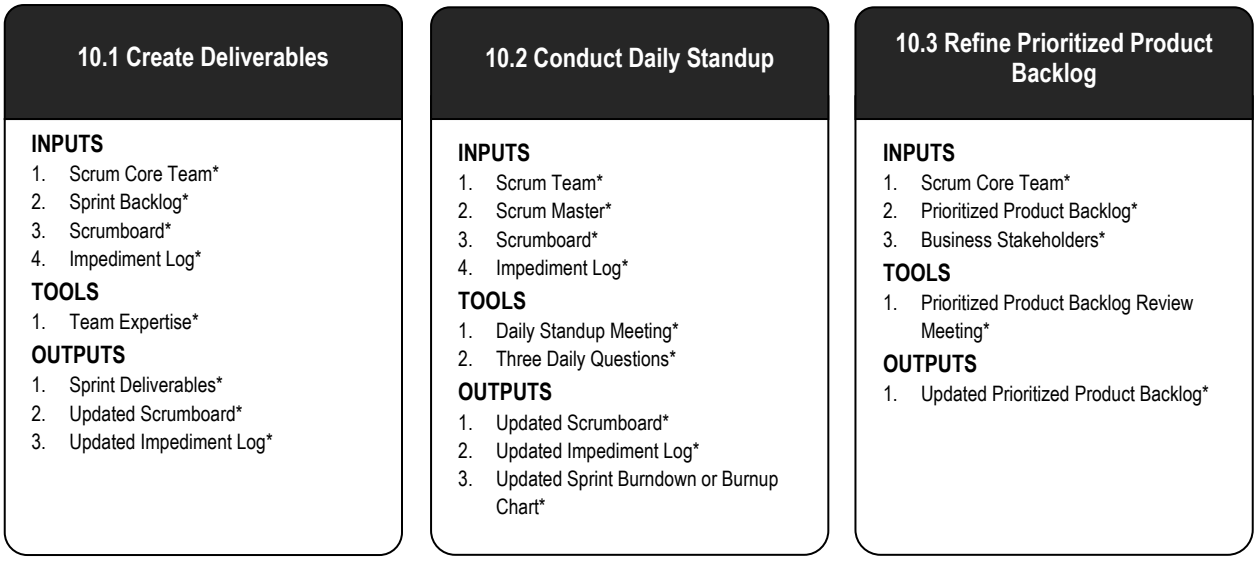

Figure 10-2: Implement Overview (Essentials)

Note: Asterisks (*) denote a "mandatory" input, tool, or output for the corresponding process.

10.1 Create Deliverables

In this process, the Scrum Team creates the Sprint deliverables by working on the tasks in the Sprint Backlog. This is the process where the Scrum Team and the Scrum Master spend most of their time. The team is supported by the Scrum Master, who facilitates meetings for the team, addresses impediments the team faces, and does whatever he/she can do to allow the Scrum Team members to focus on the creation of the Sprint Deliverables.

The Scrum Team uses a Scrumboard to track its progress during the Sprint. The Scrum Team uses the information about its progress to get a good indication of its ability to deliver according to its commitment and, if necessary, to take action to secure the most valuable outcome of the Sprint that is possible under the given circumstances.

Figure 10-3 shows all the inputs, tools, and outputs for the *Create Deliverables* process.

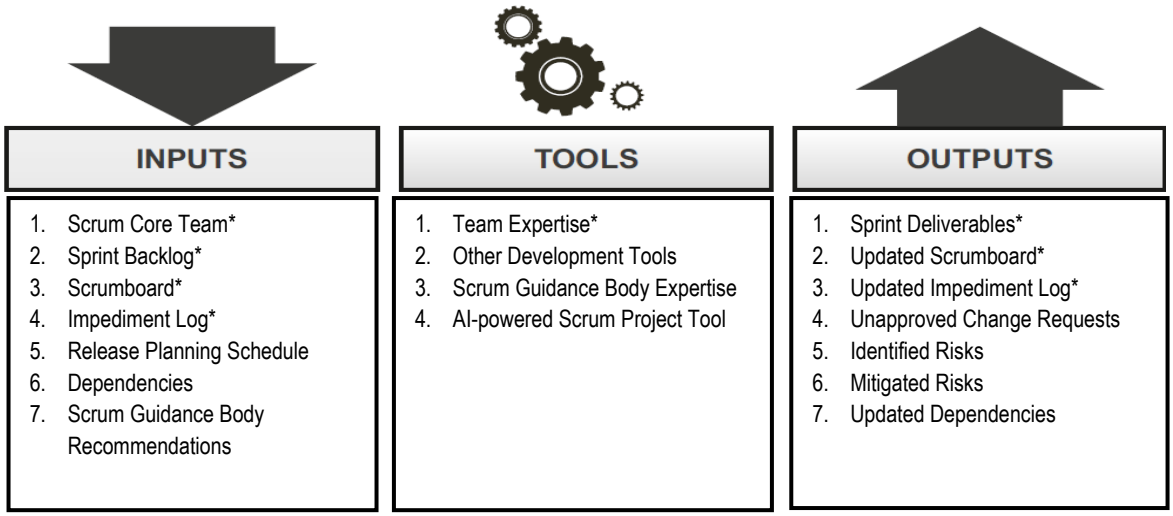

Figure 10-3: Create Deliverables—Inputs, Tools, and Outputs

Note: Asterisks (*) denote a "mandatory" input, tool, or output for the corresponding process.

Figure 10-4 a data flow diagram for “create deliverables.” It illustrates how scrum team input and planning artifacts flow into tools and processes to result in deliverables and related updates.

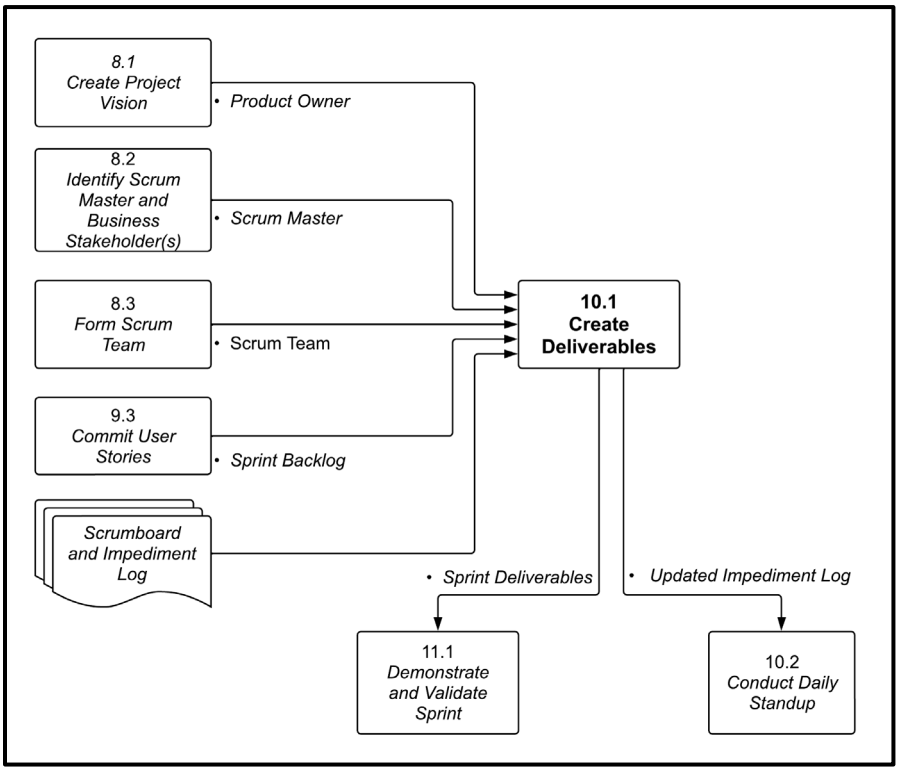

Figure 10-4: Create Deliverables—Data Flow Diagram

10.1.1 Inputs

10.1.1.1 Scrum Core Team*

Described in section 3.2.1.

10.1.1.2 Sprint Backlog*

Described in sections 9.3.3.2 and 9.6.3.1.

10.1.1.3 Scrumboard*

Described in sections 9.3.3.3 and 9.4.3.2.

Scrum's transparency comes from openly viewable information tools like the Scrumboard, which shows the team's progress. The team uses a Scrumboard to plan and track progress during each Sprint. The Scrum Team should update the Scrumboard as required so that the Scrumboard provides accurate visual information and control about the work going on as agreed and committed by the team.

The team uses the Scrumboard created from the Sprint Backlog in the Plan and Estimate Phase, which would initially have all the tasks in the "To Do" column at the beginning of the Sprint. Scrum Team members review the User Stories and Tasks in the Scrumboard on a daily basis and keep moving tasks to "In Progress" and "Complete" columns as the work progresses. An additional column (such as "Testing") may be added to the Scrumboard depending on the workflow of the Scrum Team as they create deliverables.

As the team keeps adding/updating tasks, and assigning tasks to work on, the Scrumboard keeps getting updated with the additional tasks, and the status of the tasks. For example, Figure 10-5 shows that all the tasks for User Story 1 are complete; but the team is currently working on some tasks for User Stories 2 and 3. User Story 4 has been decomposed into tasks, but the Scrum Team has not yet started working on the tasks for this User Story.

Figure 10-5 shows a scrumboard with columns for to do, in progress, and complete. Reflects real-time team progress and helps track task status throughout the sprint cycle.

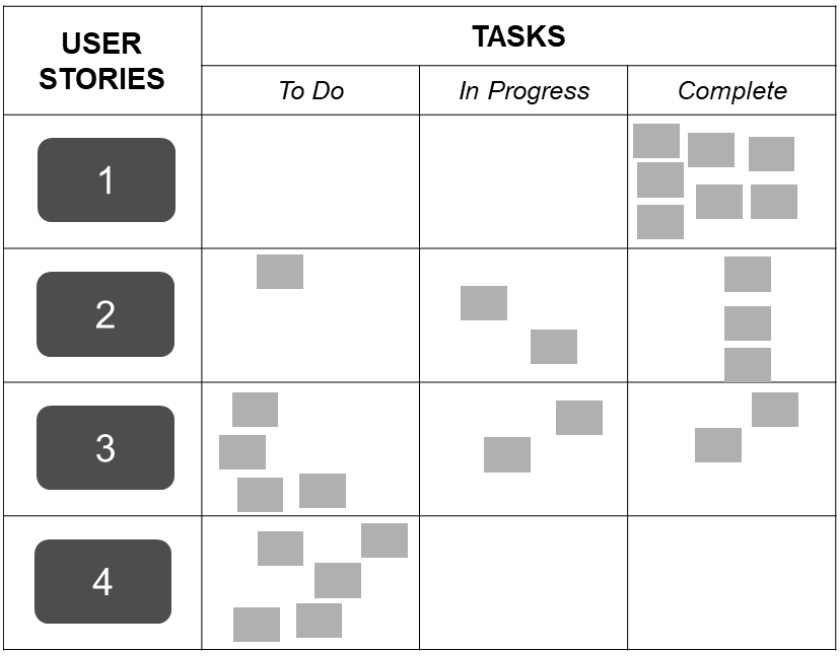

Figure 10-5: Scrumboard with Tasks To Do, In Progress, and Complete

To ensure that Scrum Team members take ownership for their work, it is recommended that the Scrum Team member working on a particular task moves the task from "To Do" to "In Progress" and puts his/her name on the task, so that the task is self-assigned to the person responsible for completing it. Also, only one Scrum Team member should be responsible for completing each task. So, User Stories should be broken down into tasks such that only one person can be responsible for one task until its completion.

As the Scrum Team starts working on a User Story, they may have gained a better understanding of the tasks required to complete the User Story. This may necessitate tasks being added, updated, or deleted from the Scrumboard as decided by the Scrum Team.

When all the tasks for a User Story are completed (e.g., as in User Story 1 in the above figure), the User Story is considered completed by the Scrum Team. Completed User Stories are then available for the Product Owner to review and either approve or reject. The review of User Stories by the Product Owner can be done by the Product Owner either, after the User Stories are completed, or during the *Demonstrate and Validate Sprint* process. If a User Story is approved by the Product Owner, then that User Story is considered as "Done" by the Scrum Team (and no more work needs to be done by the Scrum Team on that User Story).

If the Product Owner rejects a User Story, the Product Owner needs to provide his or her inputs about why the User Story was rejected (i.e., which elements of the Acceptance Criteria and/or Done Criteria were not met). Depending on the time remaining in the Sprint, after a User Story is rejected, and the reasons of the Product Owner for rejecting the User Story are provided, there will be two options available to the Scrum Team:

- Work on the rejected User Story in the current Sprint (based on inputs provided by the Product Owner) and then when all tasks required for the User Story are completed; the Scrum Team can re-submit the User Story to the Product Owner for approval during the same Sprint.
- Do not work on the rejected User Story in the same Sprint. In this case the User Story goes back into the Prioritized Product Backlog so that it can be assigned to another Sprint. The User Story may be assigned again to the same Scrum Team, or another Scrum Team may become responsible for that User Story in a future Sprint.

The Scrumboard can be maintained manually on paper or on a large whiteboard, but it can also be maintained electronically in a spreadsheet or using an AI-powered Scrum Project Tool. One Scrumboard is valid for the duration of one Sprint. The Scrum Team will create a new Scrumboard in the next Sprint.

10.1.1.4 Impediment Log*

An impediment is any hindrance or hurdle that reduces the productivity of the Scrum Team. Impediments must be identified, resolved, and removed if the team is to continue working effectively. Impediments can be internal to the team, such as inefficient workflow or lack of communication, or they can be external. Examples of external impediments might include software license issues or unnecessary documentation requirements. The Scrum framework, with its inherent transparency, facilitates the swift and easy identification of impediments. Failure to identify or deal with impediments can be very costly. Impediments should be formally recorded by the Scrum Master in an Impediment Log and should be discussed during Daily Standup Meetings and Sprint Review Meetings as appropriate.

10.1.1.5 Release Planning Schedule

Described in section 8.6.3.1.

10.1.1.6 Dependencies

Described in section 9.4.3.4.

10.1.1.7 Scrum Guidance Body Recommendations

In the *Create Deliverables* process, Scrum Guidance Body Recommendations may include best practices to effectively create deliverables, including preferred methods to conduct reviews, perform testing, create documentation, and so on. For more information on Scrum Guidance Body Recommendations, see section 8.1.1.7.

10.1.2 Tools

10.1.2.1 Team Expertise*

This refers to the collective expertise of the Scrum Team members in understanding 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. This judgment and expertise are applied to all technical and management aspects of the project during the *Create Deliverables* process. Scrum Team members have the authority and responsibility to determine the best means for converting the Prioritized Product Backlog Items into finished deliverables or increments, without requiring the involvement of any business stakeholders outside the team. Additional expertise is available from the Scrum Guidance Body, as required.

10.1.2.2 Other Development Tools

Based on the specific requirements of the project and on industry specifications, other development tools can be used accordingly. Some examples are as follows:

1. **Refactoring**

   Refactoring is a technique 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

Regular refactoring optimizes code design a little at a time, over a period of time. Refactoring results in cleaner, more maintainable code, while preserving all functionalities.

2. **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.

10.1.2.3 Scrum Guidance Body Expertise

In *Create Deliverables* processes, Scrum Guidance Body Expertise could relate to documented rules and regulations development guidelines; and/or standards and best practices for creating the deliverables (e.g., guidance on how to conduct reviews or testing). There may also be a team of subject matter experts available who can provide guidance to the Scrum Team when creating the deliverables. This team could include lead architects, senior developers, security experts, or other experienced people. For more information on Scrum Guidance Body Expertise, see section 8.4.2.7.

10.1.2.4 AI-powered Scrum Project Tool

Described in section 2.4.4

10.1.3 Outputs

10.1.3.1 Sprint Deliverables*

At the end of each Sprint, a product increment or deliverable is completed. The deliverable should possess all features and functionality defined in the User Stories included in the Sprint and should have been tested successfully.

10.1.3.2 Updated Scrumboard*

The Scrumboard is updated regularly as the team completes tasks. However, at the end of the Sprint, the Scrumboard will be reset or wiped off and a new Scrumboard is created for the next Sprint. For more information on the Scrumboard, see sections 9.3.3.3 and 9.4.3.2.

10.1.3.3 Updated Impediment Log*

Described in section 10.1.1.4.

10.1.3.4 Unapproved Change Requests

Described in section 8.4.1.5.

10.1.3.5 Identified Risks

Described in section 8.4.3.4.

10.1.3.6 Mitigated Risks

As the Scrum Team executes the work of creating deliverables (according to the User Stories in the Prioritized Product Backlog), they carry out the mitigation actions that were defined to address any previously identified risks. Throughout the *Create Deliverables* process, the team documents any newly identified risks and mitigating actions taken. The record of project risks is a living document, continuously updated throughout the project by the team to reflect the current status of all risks. Additional information about managing risks is described in section 7.4.4.

10.1.3.7 Updated Dependencies

Described in section 8.5.2.6.

10.2 Conduct Daily Standup

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

Figure 10-6 shows all the inputs, tools, and outputs for the *Conduct Daily Standup* process.

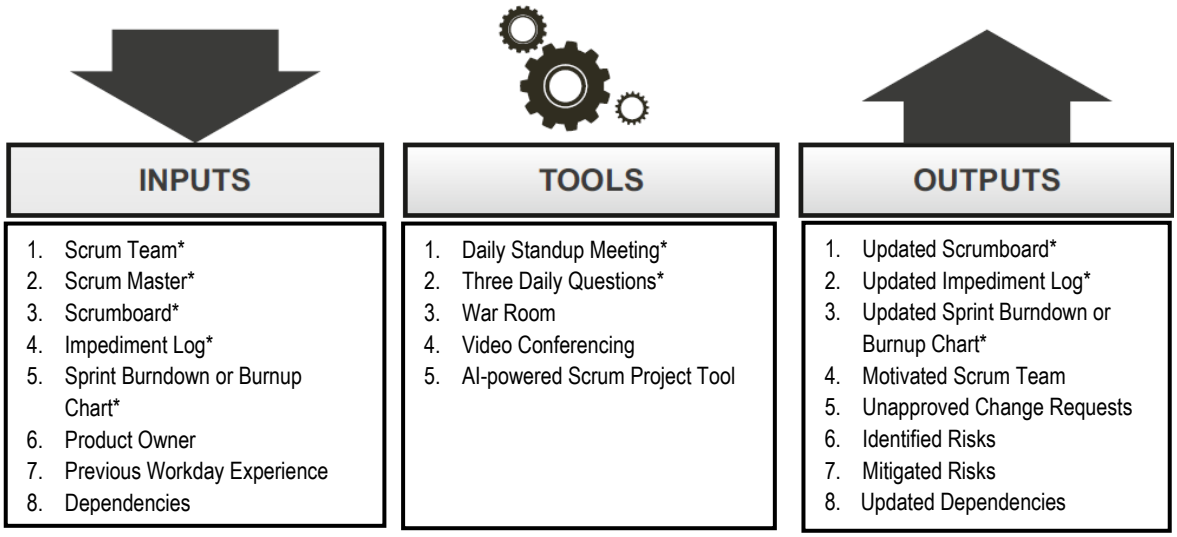

Figure 10-6: Conduct Daily Standup—Inputs, Tools, and Outputs

Note: Asterisks (*) denote a "mandatory" input, tool, or output for the corresponding process.

Figure 10-7 is a data flow diagram for the “conduct daily standup” process. Displays how daily inputs feed into tools like standup meetings and lead to updated logs, charts, and team motivation.

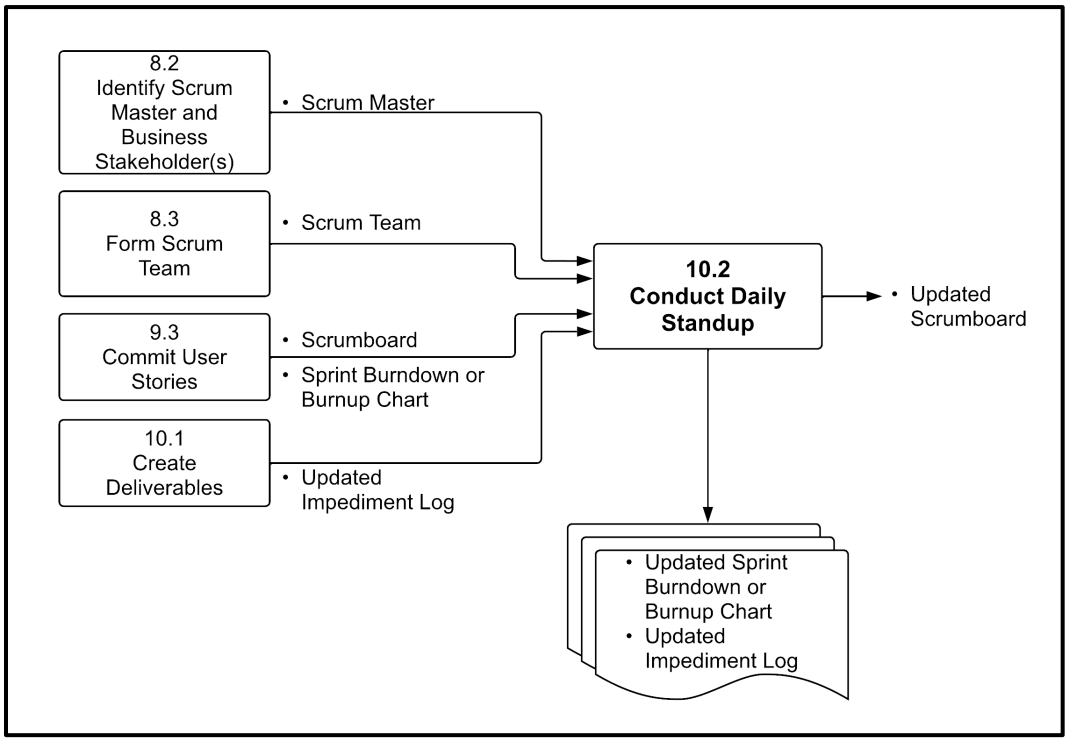

Figure 10-7: Conduct Daily Standup—Data Flow Diagram

10.2.1 Inputs

10.2.1.1 Scrum Team*

Described in section 8.3.3.1.

10.2.1.2 Scrum Master*

Described in section 8.2.3.1.

10.2.1.3 Scrumboard*

Described in sections 9.3.3.3, 9.4.3.2, and 10.1.1.3.

10.2.1.4 Impediment Log*

Described in section 10.1.1.4.

10.2.1.5 Sprint Burndown or Burnup Chart

Described in section 9.6.3.3.

10.2.1.6 Product Owner

Described in section 8.1.3.1.

10.2.1.7 Previous Work Day Experience

The Scrum Team members give status updates to fellow team members in the Daily Standup Meeting. This session is called standup because members stand throughout the meeting. Team members formulate their achievements and experiences from the previous workday. This experience is an important input to the Daily Standup Meeting.

10.2.1.8 Dependencies

Described in section 9.4.3.4.

10.2.2 Tools

10.2.2.1 Daily Standup Meeting*

The Daily Standup Meeting is a short daily meeting, Time-boxed to 15 minutes. Team members assemble to report their previous day's progress in the Sprint and plan the current day's activities. The meeting duration is intentionally short (which standing up at the meetings helps to reiterate) and all members of the Scrum Team are expected to attend. The meeting should not be cancelled or delayed if one or more team members are not able to attend. The Scrum Team manages the Daily Standup Meeting while the Scrum Master facilitates the meeting, as needed.

In the meeting, each Scrum Team member provides answers to the three daily questions (see section 10.2.2.2). Discussions between the Scrum Master and the team or between some Scrum Team members are encouraged, but such discussions happen after the meeting to ensure that the Daily Standup Meeting is kept short.

10.2.2.2 Three Daily Questions*

In the Daily Standup Meeting facilitated by the Scrum Master, each Scrum Team member provides information in the form of answers to the following three specific 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?

By focusing on these three questions, the entire team can have a clear understanding of the work status for the current Sprint. Occasionally, other items may be discussed, but this is kept to a minimum in light of the Time-boxed nature of the meeting.

It is highly recommended that the first two questions should be answered by team members in a quantifiable manner when possible, instead of qualitative lengthy answers. Team members can organize additional meetings after the Daily Standup Meeting to address items that need additional discussion.

10.2.2.3 War Room

When applying Scrum practices on a project, it is preferable for the team to be colocated, with all team members working at the same location. The term commonly used to describe this place is the War Room. Normally, the room 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. Typically index cards, sticky notes, and other low-tech, high-touch tools are made available in the room to facilitate workflow, collaboration, and problem solving.

The room is sometimes noisy due to team conversations, but these conversations contribute to the team's progress. A good War Room is cubicle free and allows the entire team to sit together ensuring face-to-face communication, which leads to team building and openness. The War Room is also ideal for conducting Daily Standup Meetings. Business stakeholder members from other Scrum Teams could also attend the War Room and discuss relevant issues, as needed.

10.2.2.4 Video Conferencing

In real-life situations, it may not always be possible for the entire Scrum Team to be colocated. In such cases, it becomes imperative to use video conferencing tools to enable face-to-face communication. For more information on effective collaboration in distributed teams, see section 2.5.3.

10.2.2.5 AI-powered Scrum Project Tool

Described in section 2.4.4

10.2.3 Outputs

10.2.3.1 Updated Scrumboard*

The Scrumboard continues to be updated regularly as the team completes tasks. For more information on the Scrumboard, see sections 9.3.3.3 and 9.4.3.2.

Figure 10-8 is a Vabro interface showing scrumboard usage in implementation. Visualizes real-time progress tracking, team assignments, and task movement across sprint phases using ai-powered features.

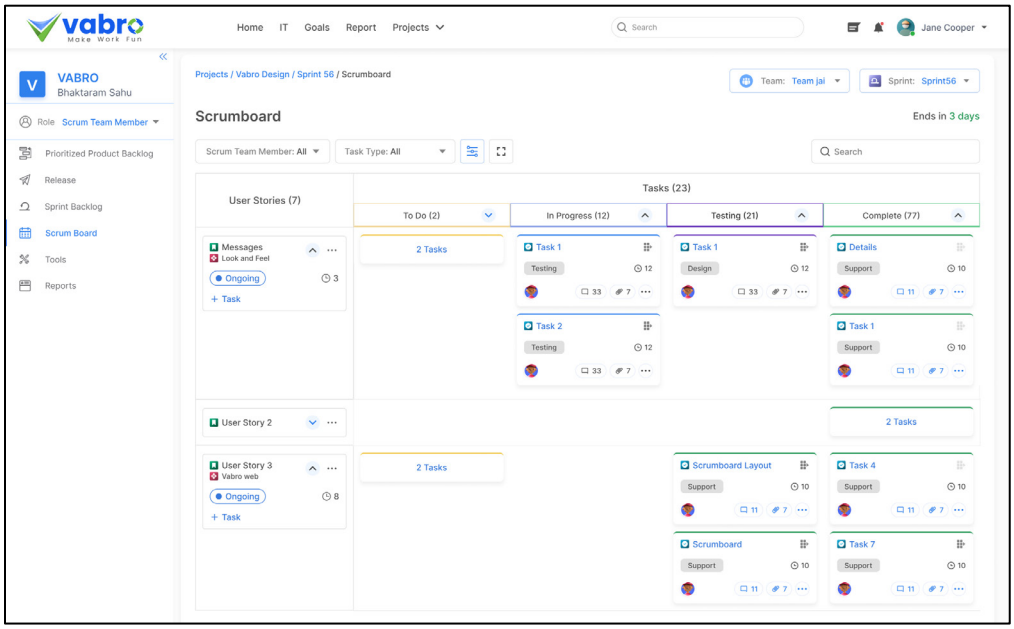

Figure 10-8: Use of Scrumboard (Source: Vabro)

Figure 10-9 is a Jira-based scrumboard showcasing its role in managing a Scrum initiative. Features task status columns, progress tracking, and backlog integration for sprint execution.

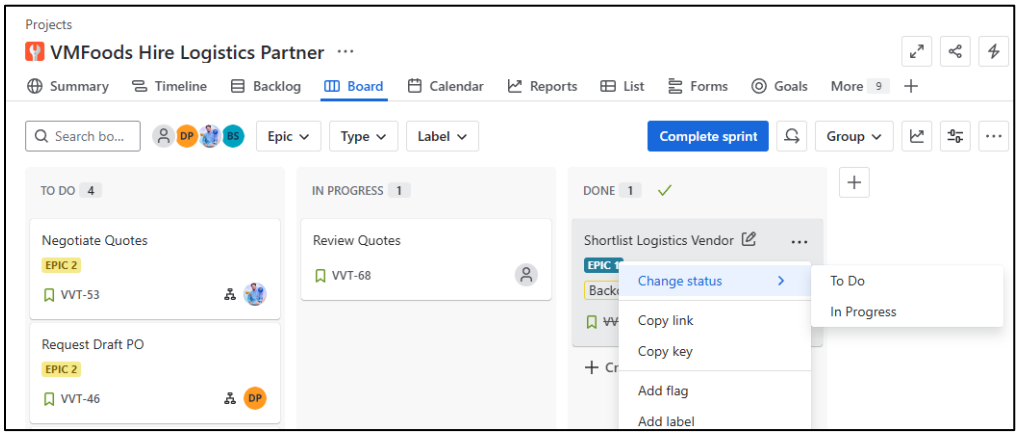

Figure 10-9: Role of Scrumboard in Scrum (Source: Jira)

10.2.3.2 Updated Impediment Log*

Described in section 10.1.1.4.

10.2.3.3 Updated Sprint Burndown or Burnup Chart

The Sprint Burndown Chart should be updated daily to show the progress that has been made by the Scrum Team and to also allow for the detection of estimates that may have been incorrect. If the Sprint Burndown Chart shows that the Scrum Team is not on track to finish the tasks in the Sprint on time, the Scrum Master should identify any obstacles or impediments to successful completion and try to remove them. For more information on the Sprint Burndown Chart, see section 9.6.3.3.

10.2.3.4 Motivated Scrum Team

The Daily Standup Meetings propagate the idea that each member of the team is important, and that he/she is a major contributor to the project. This can help improve individual and team morale. The practice of self-organizing teams can also help improve overall motivation, lead to enhanced performance of the team, and improve the quality of the deliverables being produced. For more information on the Scrum Team, see section 8.3.3.1.

10.2.3.5 Unapproved Change Requests

Described in section 8.4.1.5.

10.2.3.6 Identified Risks

Described in section 8.4.3.4.

10.2.3.7 Mitigated Risks

Described in section 10.1.3.6.

10.2.3.8 Updated Dependencies

Described in section 8.5.2.6.

10.3 Refine Prioritized Product Backlog

In this process, the Product Owner continuously updates and maintains the Prioritized Product Backlog. A Prioritized Product Backlog Review Meeting may be held, during which any changes or updates to the Product Backlog are discussed and incorporated into the Prioritized Product Backlog as appropriate.

In order to keep the Prioritized Product Backlog up to date with any change in requirements and/or priorities, the Product Owner continually works with the customer and other business stakeholders to capture and understand any changes in their needs.

To ensure that the set of User Stories the Product Owner would like the team to commit to in the next Sprint will be ready for commitment, the Product Owner refines existing Epics and User Stories in the Prioritized Product Backlog, and ensures that the User Stories satisfy the Definition of Ready.

As part of refining the Prioritized Product Backlog, the Product Owner also works with the Scrum Team to get feedback and questions related to the updates in the Prioritized Product Backlog, potentially including estimates. If changes in requirements and/or the overall progress of the Scrum Team require changes to the Release Schedule and/or the business justification, the Product Owner will also make these changes during this process. This is the process where the Product Owner will spend most of his/her time.

Figure 10-10 shows all the inputs, tools, and outputs for the *Refine Prioritized Product Backlog* process.

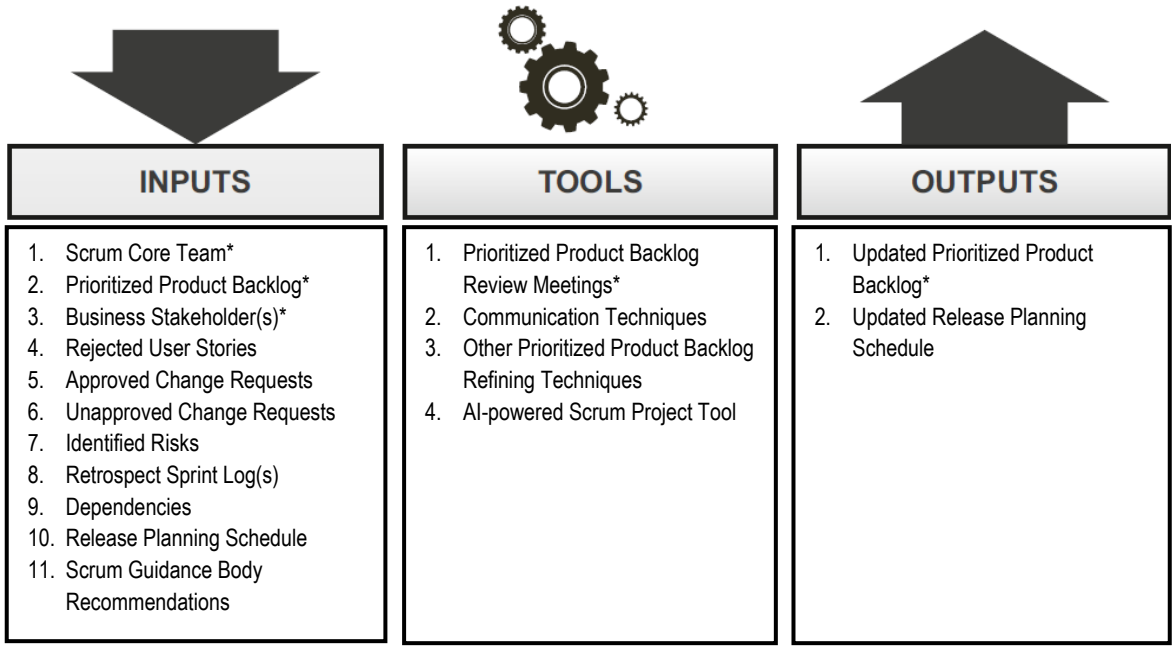

Figure 10-10: Refine Prioritized Product Backlog—Inputs, Tools, and Outputs

Note: Asterisks (*) denote a "mandatory" input, tool, or output for the corresponding process.

Figure 10-11 is a data flow diagram for the “Refine Prioritized Product Backlog” process. Tracks how changes in user stories and backlog items flow through refinement to updated backlog versions.

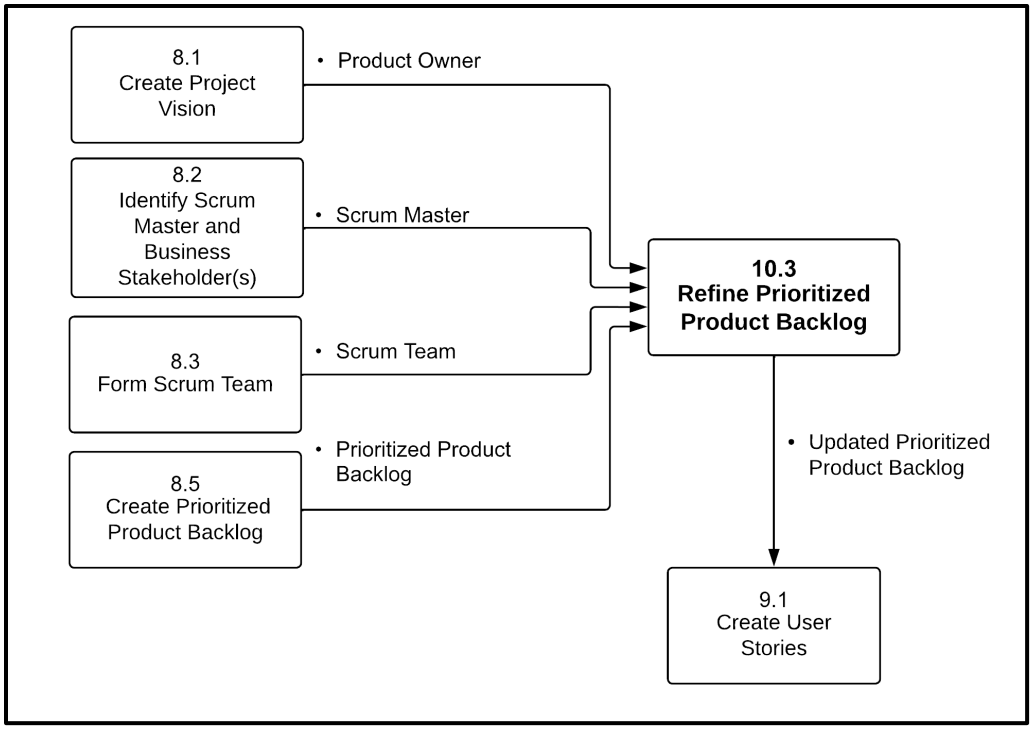

Figure 10-11: Refine Prioritized Product Backlog—Data Flow Diagram

10.3.1 Inputs

10.3.1.1 Scrum Core Team*

Described in section 3.2.1.

10.3.1.2 Prioritized Product Backlog*

Described in section 8.5.3.1.

10.3.1.3 Business Stakeholder(s)*

In order to keep the Prioritized Product Backlog up to date with any change in requirements and/or priorities, the Product Owner continually works with the customer and other business stakeholders to capture and understand any changes in their needs. For more information on business stakeholders, see section 3.2.2.

10.3.1.4 Rejected User Stories

In cases where a User Story does not meet the Acceptance Criteria, it is considered a rejected user story. The rejected user stories are normally not kept in a separate list. They simply remain in the Prioritized Product Backlog similar to other user stories in the backlog, and do not get marked as done so that they can be reprioritized in the *Refine Prioritized Product Backlog* process and be considered for development in the next or other future Sprint.

10.3.1.5 Approved Change Requests

Described in section 8.4.1.4.

10.3.1.6 Unapproved Change Requests

Described in section 8.4.1.5.

10.3.1.7 Identified Risks

Described in section 8.4.3.4.

10.3.1.8 Retrospect Sprint Log(s)

Described in section 11.2.3.4.

10.3.1.9 Dependencies

Described in section 9.4.3.4.

10.3.1.10 Release Planning Schedule

Described in section 8.6.3.1.

10.3.1.11 Scrum Guidance Body Recommendations

In the *Refine Prioritized Product Backlog* process, recommendations from the Scrum Guidance Body may include best practices on how to systematically understand and collate requirements from business stakeholder(s) and Scrum Teams and then properly prioritize the Product Backlog and finally communicate updates to all relevant persons involved with the Scrum project. For more information on Scrum Guidance Body Recommendations, see section 8.1.1.7.

10.3.2 Tools

10.3.2.1 Prioritized Product Backlog Review Meetings*

The Product Owner may have separate meetings with relevant business stakeholder(s), the Scrum Master, and the Scrum Team to ensure that he or she has enough information to appropriately update the Prioritized Product Backlog during the *Refine Prioritized Product Backlog* process. The intent of these Prioritized Product Backlog Review Meetings is to ensure that the User Stories and their corresponding Acceptance Criteria are understood, and are written properly by the Product Owner so that they reflect the stated customer requirements and priorities; User Stories are understood by everyone in the Scrum Team; and that high-priority User Stories are well-refined so that the Scrum Team can properly estimate and commit to such User Stories. The Prioritized Product Backlog Review Meetings also ensure that irrelevant User Stories are removed and any Approved Change Requests or identified risks are incorporated into the Prioritized Product Backlog.

10.3.2.2 Communication Techniques

Scrum principles and practices promote accurate and effective communication primarily through colocation of the Scrum Team. Scrum also favors informal, face-to-face interactions over formal written communications. When a Scrum Team needs to be distributed, the Scrum Master should ensure that effective communication techniques and perhaps an AI-powered Scrum Project Tool are available for use so that distributed teams can self-organize and work effectively. For more information on distributed teams, see section 2.5.3.

10.3.2.3 Other Prioritized Product Backlog Refining Techniques

Some other Prioritized Product Backlog refining tools include many of the same tools used for the following processes:

- *Develop Epic(s)*—Described in section 8.4.2.
- *Create Prioritized Product Backlog*—Described in section 8.5.2.
- *Conduct Release Planning*—Described in section 8.6.2.
- *Create User Stories*—Described in section 9.1.2.

- *Estimate User Stories*—Described in section 9.2.2.
- *Commit User Stories*—Described in section 9.3.2.
- *Identify Tasks*—Described in section 9.4.2.
- *Estimate Tasks*—Described in section 9.5.2.

10.3.2.4 AI-powered Scrum Project Tool

Described in section 2.4.4

Figure 10-12 shows Vabro interface for product backlog refinement. Features include item scoring, stakeholder feedback, and ai-generated suggestions to improve backlog prioritization.

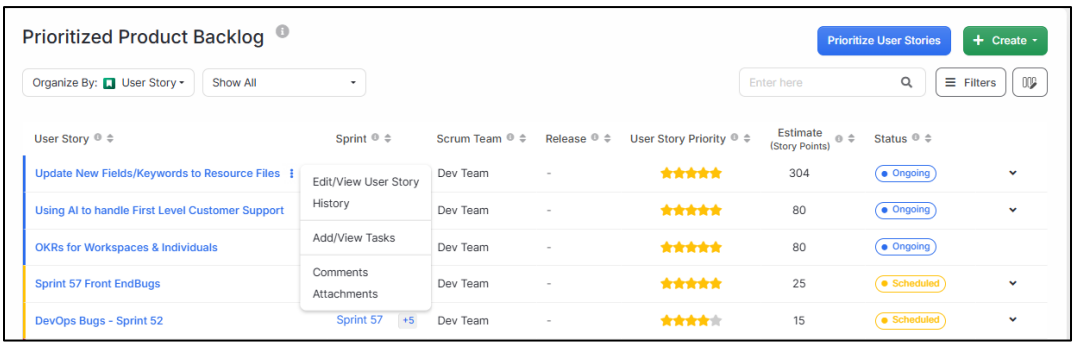

Figure 10-12: Refining Product Backlog in Scrum (Source: Vabro)

Figure 10-13 is a Jira interface for backlog refinement. Visual elements support backlog item editing, prioritization adjustments, and collaboration across scrum roles.

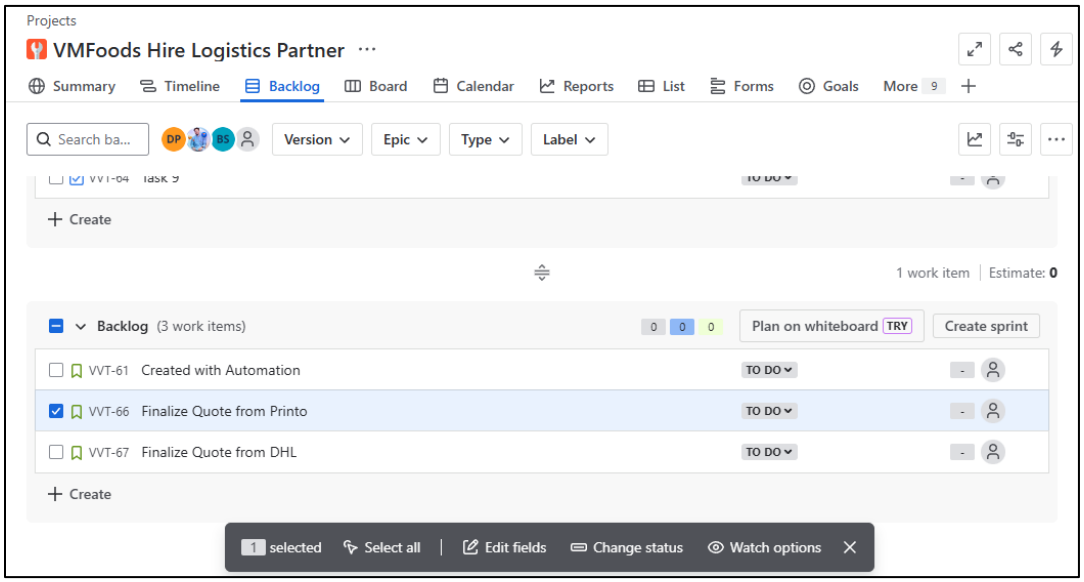

Figure 10-13: Product Backlog Refinement in Scrum (Source: Jira)

10.3.3 Outputs

10.3.3.1 Updated Prioritized Product Backlog*

The Prioritized Product Backlog may be updated with new or updated User Stories; work related to new Change Requests or identified risks; or to reflect the reprioritization of existing User Stories. For more information on the Prioritized Product Backlog, see section 8.5.3.1.

10.3.3.2 Updated Release Planning Schedule

The Release Planning Schedule may be updated to reflect the impact of new or changed User Stories in the Prioritized Product Backlog. For more information on the Release Planning Schedule, see section 8.6.3.1.

10.4 Implement Phase Data Flow Diagram

Figure 10-14 is a comprehensive data flow diagram representing the entire “implement” phase. Maps how each process and tool interact, from planning to sprint execution, to produce working deliverables and updated scrum artifacts.

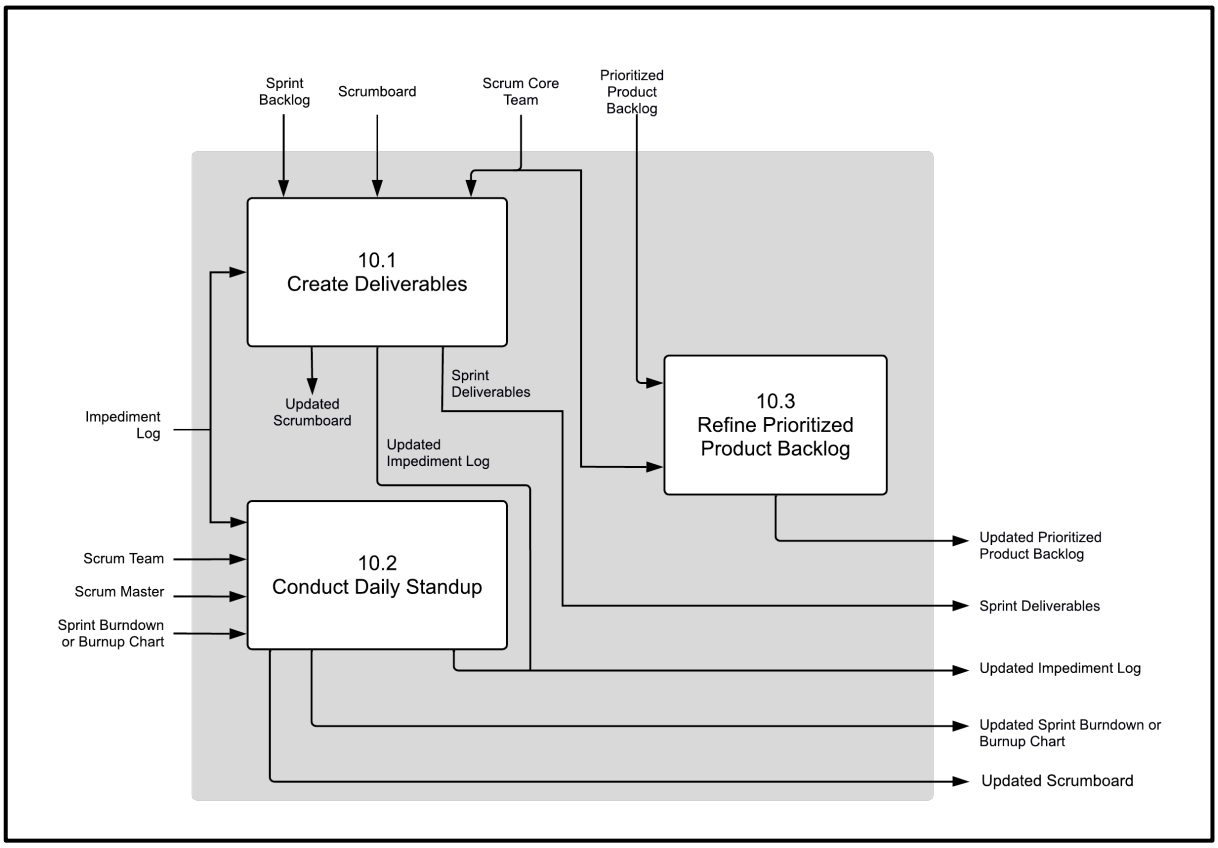

Figure 10-14: Implement Phase—Data Flow Diagram

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.