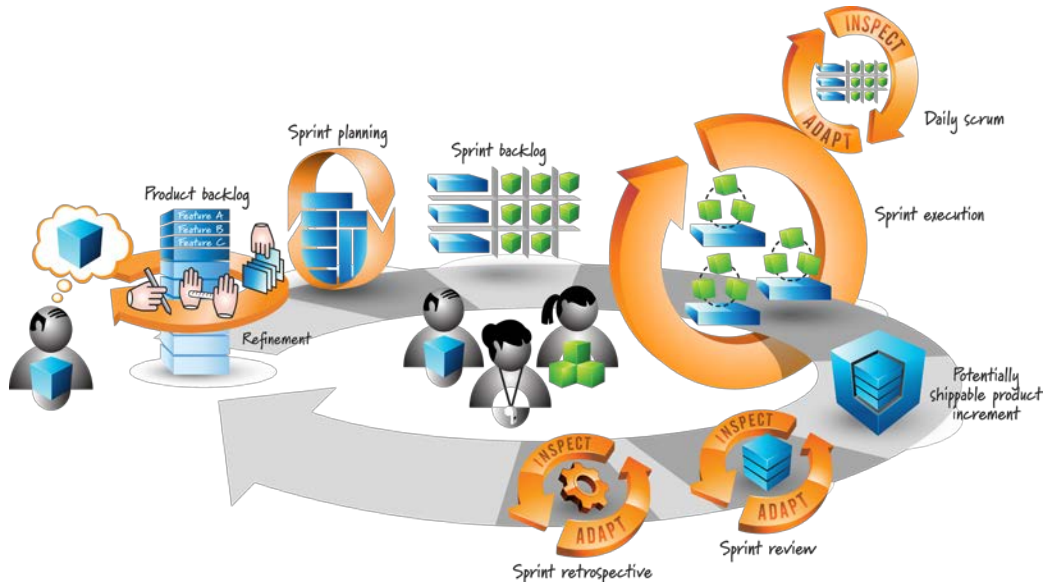


TEAM LEVEL AGILE DEVELOPMENT

— Version 1.0 —



Basic Agile Policies & Procedures

— For General Workflows —

No.	Team Level Agile Development - Policy & Procedure
1	Iteration Planning
2	Backlog Refinement
3	Team Sync
4	Iteration Review
5	Iteration Retrospective

TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

1.0 ITERATION PLANNING

1.1 PURPOSE

The purpose of iteration planning is to define, organize, and commit to the work the team will do for the iteration. Most teams timebox an hour for this meeting, but new teams may need longer as they create a working pattern. Scrum masters and team coaches typically facilitate iteration planning for the team, but product owners (POs) may sometimes lead this event.

1.2 REVISION HISTORY

Author	Description	Initials

1.3 PERSONS AFFECTED

Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary).

1.4 POLICY

1.4.1 The policy of this organization is to ensure. Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Iteration Planning events (in order to develop sprint plans and work items).

1.5 RESPONSIBILITIES

1.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Closing out the previous iteration, Establishing team capacity, Analyzing and estimating stories, Developing iteration goals, and Committing to iteration goals.

1.6 INPUTS

1.6.1 Set up a regular cadence to hold iteration planning. Make sure this meeting is held at a time everyone on the team knows about and can attend.

1.6.2 Check the team's backlog. Make sure there are sufficient stories to pull from for the upcoming iteration.

1.6.3 If using a team capacity tracker. Ensure that it accounts for any known holidays or days off that team members have submitted so that capacity for the iteration is current.

1.6.4 Review the team PI objectives committed to in PI planning. Consider what features are planned for the upcoming iteration so you can work with the team on prioritizing stories that will support feature releases and PI objectives.

1.7 PROCEDURE

1.7.1 [5 minutes] Closing out the previous iteration

Make sure that all the stories the team had been working on in the previous iteration are moved to the completed or accepted column of the SAFe team Kanban. If any stories remain, plan for what iteration they may move to and who will work on them.

1.7.2 [5 minutes] Establishing team capacity

The team can quantify the time they have to perform work in this iteration by having each team member acknowledge any time off or unavailable time for other duties. Some teams have a shared capacity tracker to help visualize this. Then, using their historical velocity as a starting point, the team makes adjustments based on the unavailable time for each team member to determine the actual capacity for the iteration.

1.7.3 [20-30 minutes] Analyzing and estimating stories

In conversation with the PO, the team selects the most valuable stories for work; typically, stories that move the team closer to meeting PI objectives and ensure dependencies with other teams are met. Each story that may be moved from the backlog to this iteration should be discussed, covering relative difficulty, size, complexity, uncertainty, technical challenges, and acceptance criteria. Planning stops once the team runs out of capacity.

1.7.4 [5-10 minutes] Developing iteration goals

The team now synthesizes the work they have planned into iteration goals that summarize the work the team plans to accomplish this iteration. Note: Some teams start this meeting with iteration goals and then work on capacity, story analysis and estimating to support those goals.

1.7.5 [5-10 minutes] **Committing to iteration goals**

Next, revisit and restate the iteration goals if needed. Then, the team commits to those goals, often with a team vote (Roman voting, fist of five, etc.).

1.8 **POST EVENT ACTIONS**

1.8.1 **Engage in followup conversations** with product owner, scrummaster, scrum team, stakeholders, other agile teams, domain experts, and other subject matter experts to clarify scope of user stories and acceptance criteria.

1.8.2 **Adjust team capacity** if any team members announce new activities (days off, etc.) that will change their capacity.

1.8.3 **Share new iteration goals** on any communication channels the organization uses to visualize goals.

1.8.4 **Make sure the team's work management tool is up to date** with the iteration plan.

1.8.5 **Plan any conversations with teams** that will do work your team is dependent on, or any teams looking to your team for work.

1.9 **OUTPUTS**

1.9.1 Stories are planned for the upcoming iteration.

1.9.2 Committed iteration goals.

1.9.3 Dependencies with other teams are confirmed/handled.

TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

2.0 BACKLOG REFINEMENT

2.1 PURPOSE

The purpose of backlog refinement is for agile teams to take some time during the iteration to plan for the future, although agile teams should be primarily focused on the current iteration goals. Backlog refinement sessions allow teams to begin building alignment and shared understanding about upcoming work, to start identifying dependencies, to inform backlog prioritization, and to begin hypothesizing how the team may address future issues. Product owners (POs) typically facilitate backlog refinement for the team (however, scrum masters are suitable facilitators, and agile teams should do it with or without a formal facilitator on the scheduled date and time).

2.2 REVISION HISTORY

Author	Description	Initials

2.3 PERSONS AFFECTED

Product Owner, Scrum Master, Agile Team, and Stakeholders (as necessary).

2.4 POLICY

2.4.1 The policy of this organization is to ensure. Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Backlog Refinement events (in order to update and refine the product backlog, clarify, split, add, and delete work items, prioritize them, and prepare for the next sprint planning event).

2.5 RESPONSIBILITIES

2.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Discussing the most important items of work for the team to clarify first, Discussing other future stories, Estimating size and prioritizing, and Splitting large stories into smaller ones as necessary.

2.6 INPUTS

- 2.6.1 Find a meeting place. Choose a physical or virtual space that fosters communication and collaboration and keep your backlog in a tool that everyone can access and review easily.
- 2.6.2 Establish a time and cadence for the backlog refinement that the team agrees upon. Decide how often to refine the team's backlog. Holding backlog refinement at least once per iteration, but you can adjust frequency and duration based on the team's needs.
- 2.6.3 Prepare the team backlog. Visit the product backlog, ensure it's up to date, make any necessary updates or refinements, and begin reprioritizing work based on accomplishments, velocity, team changes, stakeholder requests, upcoming milestones, deadlines, system demos, releases, deployments, integration, and test events, etc.
- 2.6.4 Prepare relevant backlog refinement data. Gather a list of features, release objectives, and dependencies with other teams.
- 2.6.5 Define candidate stories for the next iteration. Begin prepopulating the product backlog with stories for the next event based on unfinished work, needed refinements and technical debt, changing priorities, and new work items needed sooner than planned.
- 2.6.6 Define team capacity allocation. Gather data on the scrum team's capacity and load for the upcoming sprint based on changing schedules, priorities, and other unplanned personal, professional, and administrative priorities.

2.7 PROCEDURE

2.7.1 Discuss the most important items of work for the team to clarify first

Often, the priority items to discuss are possible stories for the upcoming iteration. The team backlog must always contain some stories that are ready for implementation without significant risk or surprise. The team should first make sure there are enough stories in the backlog for the next iteration and validate that there are no outstanding questions or dependencies. You may spend most of your time in most of your backlog refinement meetings creating stories and acceptance criteria for the upcoming iteration.

2.7.2 Discuss other future stories

Next, the team can consider stories for future iterations. What is needed to release the features this team is working on, or to reach PI objectives? What maintenance or enablers may be needed? What work is needed to address dependencies with other teams? This is a good time to brainstorm candidate stories for the future. Importantly, the PO can get information from across the team on the scope and effort of future work to create a predictable roadmap.

2.7.3 Estimate size and prioritize

Although stories will be sized as they move from the backlog to an iteration plan, initial sizing of candidate stories can be helpful. Teams can discuss each story using the INVEST criteria (i.e., Independent, Negotiable, Valuable, Estimable, Small, and Testable) and the 3 C's method (Card, Conversation, and Confirmation), and then rank priority of all stories.

2.7.4 Split large stories into smaller ones as necessary

When stories are sized at eight points or more, have your team work on ways to split this larger story into several smaller stories. A good technique is to divide acceptance criteria into multiple stories and split those into parts as necessary. Note. Even eight-point stories are a bit large as agile teams will take an entire sprint for small stories.

2.8 POST EVENT ACTIONS

2.8.1 Update the team work management tool and team boards with any new story timing, acceptance criteria, sizing, or other details for the stories the team worked on.

2.8.2 Communicate with teams you have planned dependent work with to align on any changes to the work.

2.9 OUTPUTS

2.9.1 Agreement on the scope and effort for the work that will support team goals in the upcoming iteration(s).

2.9.2 Shared understanding of how upcoming work may be solutioned.

2.9.3 Estimated stories with acceptance criteria ready for the next iteration

2.9.4 Spikes for further investigation.

2.9.5 Identified outcomes are resolved, follow-up actions are determined, and current plan risks and impediments have been surfaced and discussed.

TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

3.0 TEAM SYNC

3.1 PURPOSE

The purpose of the team sync is to coordinate the team's activities and raise blocking issues and dependencies, some of which will need to be resolved afterward. High-performing teams use the team sync to find opportunities to help each other so that the entire team succeeds in delivering its committed iteration goals. Scrum masters and team coaches typically facilitate the team sync (however, agile teams should self-organize and execute the team sync with or without a facilitator).

3.2 REVISION HISTORY

Author	Description	Initials

3.3 PERSONS AFFECTED

Product Owner, Scrum Master, Agile Team, and Stakeholders (as necessary).

3.4 POLICY

3.4.1 The policy of this organization is to ensure. Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Team Sync events (in order to review the Scrum team's progress, track progress on a daily basis, coordinate dependencies, resolve impediments, ensure the Scrum team is operating at optimal velocity, and focus the team on accomplishing the Sprint goals or objectives).

3.5 RESPONSIBILITIES

3.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Arriving early, Starting on time, Giving individual updates, Asking brief clarifying questions, Clarifying any action items, Facilitating further discussion and problem solving, and Concluding meeting.

3.6 INPUTS

3.6.1 Select an optimal time. Establish a time for the team sync that the team agrees upon.

3.6.2 Prepare meeting space. Find a physical or virtual space which fosters communication and collaboration.

3.6.3 Prepare to move efficiently and effectively. Ensure there is time reserved for meet-afters, if needed.

3.6.4 Prepare inputs. These include updates on previously raised risks and impediments, status of iteration goals, updates visualized on work management tool or team Kanban board, updates from dependent teams, etc.

3.7 PROCEDURE

3.7.1 Arrive early

You should be ready to host and start the meeting. Team members should arrive just a bit early to be ready to participate on time.

3.7.2 Start on time

Signal that it is time to start the meeting.

3.7.3 Team members give individual updates

These should be one minute or less per person. Coach team members to give a brief update. One format that can work is answering these questions:

1. What did you do yesterday to advance the team's iteration goals?

2. What will you complete today to advance the team's iteration goals?

3. Are there any blockers or impediments for your work or for the team to complete iteration goals?

3.7.4 You and other team members should ask brief clarifying questions

Anything requiring deeper discussion or a lengthy back-and-forth is a candidate to be named in the team sync, but resolved in a meet-after or other meeting outside of the team sync.

3.7.5 Clarify any action items

You should read out any topics that require further discussion.

3.7.6 Facilitate further discussion and problem solving

Help team members organize meet-afters to address them. Meet-afters should hold only the involved people so that others can leave.

3.7.7 Conclude meeting

End the meeting on time and move into meet-afters. One technique to move through meet-afters is to start with the items that had the most participants and progressively move to those with the least participants so that team members can leave the meet-after portion efficiently.

3.8 POST EVENT ACTIONS

3.8.1 Scrum master or team coach follows up on risks and impediments.

3.8.2 Product owner updates iteration goals, if required.

3.8.3 Communicate possible impacts to dependent teams or PI objectives.

3.8.4 Update work management tool or team Kanban board with progress, impediments, notes, etc.

3.9 OUTPUTS

3.9.1 Alignment on the work completed previously and for the day ahead.

3.9.2 Surfaced risks/impediments that are blocking the team.

3.9.3 Resolution of meet-after topics or follow-up actions agreed upon.

TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

4.0 ITERATION REVIEW

4.1 PURPOSE

The purpose of the iteration review is to review the outcome of the closing iteration by showing working stories to the product owner (PO) and other stakeholders to get their feedback on progress toward the product goal. It is facilitated by the scrum master or team coach, with team members presenting their own work in the demo. It closes the iteration, demonstrates work done, generates early feedback, and provides input for upcoming iterations.

4.2 REVISION HISTORY

Author	Description	Initials

4.3 PERSONS AFFECTED

Product Owner, Scrum Master, Agile Team, and Stakeholders (as necessary).

4.4 POLICY

4.4.1 The policy of this organization is to ensure. Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Iteration Review events (in order to demonstrate measurable progress towards iteration goals, solicit stakeholder feedback, and influence future course corrections, backlog refinement, and iteration plans).

4.5 RESPONSIBILITIES

4.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Reviewing team goals, Demonstrating completed stories, Reflecting on any uncompleted goals and stories, and Refining the team backlog if needed.

4.6 INPUTS

4.6.1 Identify demos and prepare work products for fast and efficient execution. If you are planning a specific theme or goal, be sure to gather your images, set up your tools, and prepare any other props to support this early.

4.6.2 Identify the iterations goals. Share updates to iteration goals and what has and has not been accomplished.

4.6.3 Coordinate who and what will be demonstrated. Work with your team to build a plan for demoing work from the iteration.

4.6.4 Determine the order of demos for the team. Ask for volunteers, order based on importance, or other size, complexity, and time-based issues to maximize impact.

4.6.5 Ensure product owners and needed stakeholders can attend. Invite any stakeholders the team has been working with to the demo in the iteration review.

4.6.6 Identify collaborative efforts and share demo. If the team collaborated with others on this work, agree ahead of time who will demo and how the demo will be shared.

4.7 PROCEDURE

4.7.1 Review team goals

Discuss the status of each iteration goal. Teams may also review PI objectives for a larger context.

4.7.2 Demo completed stories

The review proceeds with a walkthrough and demonstration of each completed story (spikes, NFRs, and any other work completed by the team). Demos should share progress towards iteration goals, product changes, test scenarios, or a prototype that represents the user's environment. Spikes can be demoed as a presentation of finding or learning. The team and stakeholders present should ask questions and provide feedback.

4.7.3 Reflect on any uncompleted goals and stories

Next, the team should reflect on missed iteration goals and stories that were not completed to identify opportunities

for future improvement. This discussion usually results in the discovery of impediments or risks, false assumptions, changing priorities, estimating inaccuracies, or over-commitment.

4.7.4 Refine the team backlog if needed

Using feedback, the team can refine their backlog to reflect any adjustments so that the backlog is up-to-date before the next iteration planning event.

4.8 POST EVENT ACTIONS

4.8.1 Update the team backlog based on feedback from the demo.

4.8.2 Reflect on the unfinished stories and record what happened. Considering unfinished work often reveals impediments or risks, false assumptions, changing priorities, estimating inaccuracies, or over-commitment.

4.8.3 Move these findings into the iteration retrospective and consider how future iterations can be better planned and executed.

4.8.4 Identify opportunities to show integrated work at the system demo.

4.8.5 Move the work not completed to the team backlog for consideration in future iterations.

4.9 OUTPUTS

4.9.1 Planned iteration goals and work items.

4.9.2 Accomplished iteration goals and work items.

4.9.3 Impediments or dependencies preventing completion of iteration goals and work items.

4.9.4 Goals, risks, dependencies, impediments, changes to the backlog, new priorities, and work items for next iteration.

4.9.5 Iteration metrics such as historical, planned, added, deleted, and actual story points and velocity. This may include metrics such as team morale, quality, customer satisfaction, etc.

TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

5.0 ITERATION RETROSPECTIVE

5.1 PURPOSE

The iteration retrospective is used by Agile teams to reflect on the iteration just completed and to derive new ideas to improve the team's process. This helps instill the concept of relentless improvement—one of the SAFe Core Values—in the individuals and the team. It also helps ensure that every iteration yields some small improvements. The team's scrum master or team coach typically facilitates the iteration retrospective.

5.2 REVISION HISTORY

Author	Description	Initials

5.3 PERSONS AFFECTED

Product Owner, Scrum Master, Agile Team, and Stakeholders (as necessary).

5.4 POLICY

5.4.1 The policy of this organization is to ensure. Product Owners, Scrum Masters, Agile Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Iteration Retrospective events (in order to identify what's working well, what's not working well, and how to improve the next iteration's performance to achieve iteration goals efficiently and effectively, optimize team performance, and maximize resulting value).

5.5 RESPONSIBILITIES

5.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Describing the retrospective's purpose and process, Applying a variety of formats to optimize idea generation, Ensuring inputs are anonymous to maximize impact and value, Timeboxing team member thoughts, Reviewing thoughts, Brainstorming actions and improvement items, and Voting and closing up discussion.

5.6 INPUTS

5.6.1 Prepare the retrospective medium in advance. This includes meeting space, physical materials such as white boards, flip charts, markers, post-it notes, or any other special canvases. This also includes electronic mediums such as collaboration tools.

5.6.2 Have the Scrum team prepare their inputs. The team will supply the inputs to this meeting.

5.6.3 Optimize the generation of Scrum team inputs. A facilitator's job is to make it easy, comfortable, and non-judgmental for the team to share what went well, any things that didn't go well, and their thoughts on ways to improve.

5.6.4 Clearly identify how the inputs will be used most efficiently and effectively. Have a plan for how the upcoming retrospective will be used, and how you will lead the team through it so that their job to add input and ideas at each stage is clear.

5.7 PROCEDURE

5.7.1 [Five minutes] Describe the retrospective's purpose and process

Introduce the goals, retrospective format, and any specific questions. Remind everyone that the goal of the iteration retrospective is to reflect on how the team did to identify small, continual improvements that the team can own.

5.7.2 Apply a variety of formats to optimize idea generation

1. **Individual** – Individually write sticky notes and then find patterns as a group

2. **Appreciation** – Note whether someone has helped you or helped the team

3. **Conceptual** – Choose one word to describe the iteration

4. **Rating** – Rate the iteration on a scale of one to five, and then brainstorm how to make the next one a five

5. **Simple** – Open a discussion and record the results under three headings: what went well, what didn't, and actions/do better next time.

5.7.3 Ensure inputs are anonymous to maximize impact and value

Use a variety of physical and electronic mediums to maximize anonymity and creativity. Eliminate the temptation to match inputs to individuals, reward popular team members, and minimize the contributions of key performers with little power and status. Don't ask "who made this stupid remark?" Ensure the facilitator is neutral to avoid the "tried that and it didn't work before mentality!"

5.7.4 [10 minutes] Timebox team member thoughts

Ask team members to write their thoughts down and add them to the chart paper or tool being used for the retrospective. At the nine-minute mark, ask team members if they need more time or to wrap up their final thoughts.

5.7.5 [10-15 minutes] Review thoughts

Have team members share the thoughts they added or have the facilitator read team additions out so everyone can consider each idea.

5.7.6 [Five minutes] Brainstorm actions and improvement items

The team may have already added some items in the "do better next time" category, but they may have more thoughts after hearing everyone's ideas.

5.7.7 [Five minutes] Vote and close up discussion

Hold a team vote on action items and "do better next time" suggestions to select or prioritize which item(s) to add to the team backlog.

5.8 POST EVENT ACTIONS

5.8.1 Make sure you and product owner (PO) discuss who will own action items that were detailed in the retrospective.

5.8.2 Find a way to track improvement items the team has suggested and agreed upon. What happens after the retro? How many action items are completed? What is their effect?

5.8.3 Help the team come up with ways to measure the success of improvement items they put into place.

5.8.4 For issues or concerns that rise to the ART level, work with the product owner (PO) to share these with the Release Train Engineer (RTE).

5.9 OUTPUTS

5.9.1 The retrospective facilitator and the team product owner (PO) should gain one or two improvement stories to add to the backlog.

5.9.2 Follow up with the PO to make sure they have what they need to add these stories for backlog refinement.

5.9.3 Offer your team a mechanism to give feedback on the techniques, themes, and questions used at the retrospective so that you can continually improve how you facilitate this event for the team.