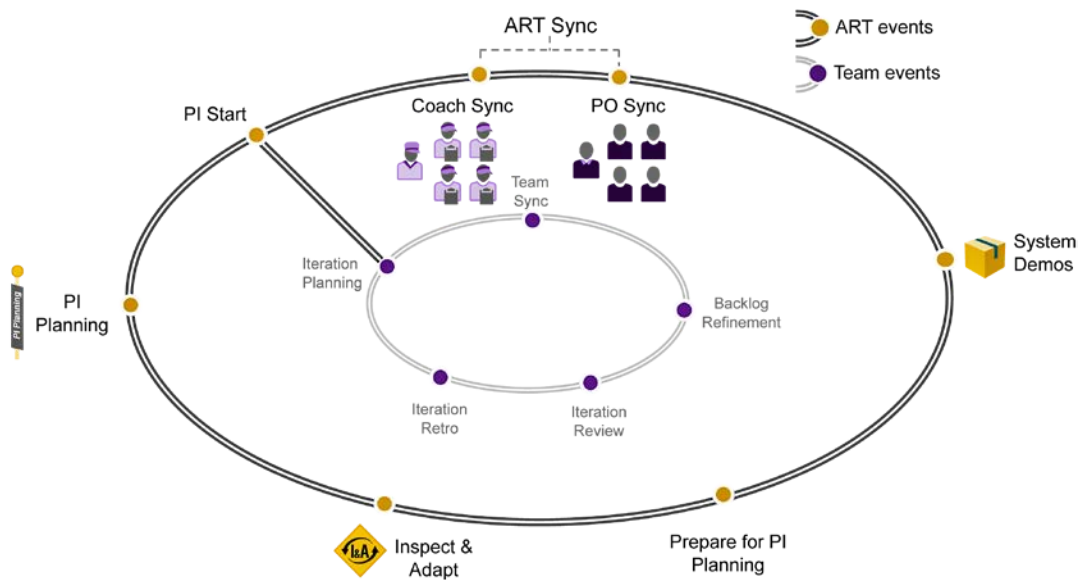


RELEASE LEVEL AGILE DEVELOPMENT

— Version 1.0 —



Intermediate Agile Policies & Procedures

— For General Workflows —

No.	Release Level Agile Development – Policy & Procedure
1	Release Planning
2	Release Refinement
3	Release Sync
4	System Demo
5	Inspect & Adapt

RELEASE LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 2 - Release

1.0 RELEASE PLANNING

1.1 PURPOSE

The purpose of Release Planning is to define, organize, and commit to the work the team of teams will do for the release (which is typically a quarter or 90-period consisting of up to five two-week iterations and a three-week innovation and planning iteration). Most teams of teams timebox two business days for this event, while smaller teams of teams may need less time, and larger teams of teams may need an extra day. Release engineers typically facilitate release planning for the team of teams, but product managers (PMs) may sometimes lead this event. Release planning is a cadence-based event for the entire team of teams that aligns agile teams and stakeholders to a shared mission and vision. Cadence-based release planning is essential for the proper health and functioning of a team of teams. Failure to apply this event undermines the ability of a team of teams to satisfy overall goals and objectives and deliver the value described by a business context. All members of a team of teams participate in release planning to align everyone to the vision, mission, goals, objectives, features, stories, risks, dependencies, and commitments of the entire team of teams to maximize visibility and transparency.

1.2 REVISION HISTORY

Author	Description	Initials

1.3 PERSONS AFFECTED

Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary).

1.4 POLICY

1.4.1 The policy of this organization is to ensure. Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Release Planning events (in order to develop release plans, release goals and objectives, identify risks and dependencies, and lower-level work items).

1.5 RESPONSIBILITIES

1.5.1 The roles and responsibilities include. Product Managers, Release Engineers, Teams of Teams are responsible for participating in sessions for Business context, Product/solution vision, Architecture vision, Planning context, Team breakouts #1, Draft plan review, Management review, Solution review, Planning adjustments, Team breakouts #2, Final plan review, Identifying risks, Confidence vote, Planning rework, Planning retrospective, Next steps, and Defining release objectives.

1.6 INPUTS

1.6.1 **Organizational Readiness.** Before release planning, there must be strategy alignment among participants, stakeholders, and Business Owners. Critical roles are assigned. To address this in advance, however, event organizers must consider the following:

1. **Planning scope and context** – Is the planning process's scope (product, system, technology domain) understood? Do we know which teams need to plan together?
2. **Business alignment** – Is there reasonable agreement on priorities among the Business Owners?
3. **Agile teams** – Do we have Agile teams? Are there dedicated team members and an identified Scrum Master/Team Coach and Product Owner for each team?

1.6.2 **Content Readiness.** It's equally important to have a clear vision and context so that the right stakeholders can participate. Therefore, the release planning must include the following:

1. **Executive briefing** – A briefing that defines the current business context.
2. **Product vision briefing(s)** – Briefings prepared by Product Management, including the top 10 features in the Release Backlog.
3. **Architecture vision briefing** – A presentation made by the CTO, Enterprise Architect, or System Architect to communicate new Enablers, features, and Nonfunctional Requirements (NFRs).

1.6.3 **Logistics Readiness.** Preparing an event to support a large number of attendees is complex. This includes securing and preparing the space for physically collocated planning. For remote attendees or a fully distributed Release Planning, this also includes investment in the necessary technical infrastructure. Considerations include:

1. **Locations** – Each location where planning takes place needs preparation in advance.
2. **Technology and tooling** – Real-time access to information and tooling to support remote distributed planning.
3. **Communication channels** – Primary and secondary audio, video, and presentation channels must be available.

1.6.4 Basic Inputs. Data from organizational, content, and logistics readiness may include:

1. **Business context** – A short speech, talk, visit, or briefing by a key organizational executive or business owner describing overall goals of the enterprise, portfolio, or value stream so that the team of teams understands what outcomes they are trying to achieve and how the solution helps achieve those outcomes.
2. **Vision and roadmap** – A brief speech, talk, visit, or briefing by a solution or product manager describing the solution or product's vision, goals, objectives, timelines, milestones, dependencies, and critical assumptions, constraints, stakeholders, and key performance indicators (KPIs) or objectives and key results (OKRs).
3. **Highest priority features in the release backlog** – A short, prioritized list of features and their key attributes aligned with the planning horizon which is generally a business quarter or 90-day period consisting of five two-week iterations and a three-week innovation and planning sprint. (Features should have been story mapped in advance to identify needed stories associated with each feature and the sprint in which they are completed).

1.7 PROCEDURE

DAY ONE

1.7.1 [60 minutes] Business context

A Business Owner or senior executive describes the current state of the business, shares the Portfolio Vision, and presents a perspective on how effectively existing solutions address current customer needs.

1.7.2 [90 minutes] Product/solution vision

Product Management presents the current vision (typically represented by the top ten or so upcoming features). They highlight changes from the previous release planning event and any relevant milestones.

1.7.3 [60 minutes] Architecture vision and development practices

The System Architect presents the architecture vision. Also, a senior development manager may introduce Agile-supportive changes to development practices, such as test automation, DevOps, Continuous Integration, and Continuous Deployment, which the teams will adopt in the upcoming release.

1.7.4 [90 minutes] Planning context and lunch

The release engineer presents the planning process and expected outcomes.

1.7.5 [3 hours] Team breakouts #1

In the breakout, teams estimate their capacity for each Iteration and identify the backlog items they will likely need to realize the features. Each team creates draft plans, visible to all, iteration by iteration.

1.7.6 [60 minutes] Draft plan review

During the tightly timeboxed draft plan review, teams present key planning outputs, which include capacity and load, draft release objectives, potential risks, and dependencies. Business Owners, Product Management, and other teams and stakeholders review and provide input.

1.7.7 [60 minutes] Management review and problem-solving

Draft plans likely present challenges like scope, people and resource constraints, and dependencies. During the problem-solving meeting, management may negotiate scope changes and resolve other problems by agreeing to various planning adjustments. The release engineer facilitates and keeps the primary stakeholders together for as long as necessary to make the decisions needed to reach achievable objectives.

1.7.8 [60 minutes] Solution review and problem solving

Solution Trains often hold an additional management review and problem-solving workshop after the first day of planning to address cross-team or team issues. Alternatively, the release engineers of the involved trains may talk with each other to discuss the problems for the team of team's specific management review and problem-solving meeting. The solution engineer helps facilitate and resolve issues across the teams of teams.

DAY TWO

1.7.9 [60 minutes] Planning adjustments

The next day, the event begins with management presenting changes to the planning scope, people, and resources.

1.7.10 [2 hours] Team breakouts #2

Teams continue planning and making the appropriate adjustments. They finalize their objectives for the release, to

which the Business Owners assign business value.

1.7.11 [60 minutes] Final plan review and lunch

All teams present their plans to the group during this session. At the end of each team's time slot, the team states its risks and impediments and provides the risks to the release engineer for use later in the ROAMing exercise. The team then asks the Business Owners if the plan is acceptable. If the plan is accepted, the team brings their team release objective sheet to the front of the room so everyone can see the aggregate objectives unfold in real-time. If the Business Owners have concerns, teams can adjust the plan to address the identified issues. The team then presents its revised plan.

1.7.12 [90 minutes] Team of teams release risks

During planning, teams have identified risks and impediments that could impact their ability to meet their objectives. These are resolved in a broader management context before the whole train. One by one, the risks are discussed and addressed with honesty and transparency and then grouped into one of the following categories:

1. **Resolved** – The teams agree that the risk is no longer a concern.
2. **Owned** – Someone on the train owns the risk since it cannot be addressed during release planning.
3. **Accepted** – Some items are simply facts or potential problems that must be understood and accepted.
4. **Mitigated** – Teams identify a plan to reduce the impact of the risk.

1.7.13 [15 minutes] Confidence vote

Once release risks have been addressed, teams vote on their confidence in meeting their team release objectives:

1. Each team conducts a vote using their fingers (fist of five) or a digital tool for remote events.
2. If the average is three fingers or above, then management should accept the commitment.
3. If it's less than three, the team reworks its plan.
4. Anyone voting two fingers or fewer should be allowed to voice their concerns.
5. These concerns might add to the risk list, require replanning, or provide information.
6. The vote is repeated for the team of teams, with everyone expressing their confidence in the collective plan.

1.7.14 [60 minutes] Plan rework

If necessary, teams adjust their objectives until they have high confidence. This additional planning is one occasion where alignment and commitment are valued more highly than adhering to a timebox.

1.7.15 [30 minutes] Planning retrospective and moving forward

Finally, the release engineer leads a brief retrospective for the release planning event to capture what went well, what didn't, and what to do better next time.

1.7.16 [15 minutes] Next steps

Typically, a discussion about the next steps, along with final instructions to the teams, follows, including:

1. Cleaning up the rooms used for planning (if applicable).
2. Entering the team release objectives and stories in Agile lifecycle management (ALM) tooling.
3. Reviewing team and team of team events calendars.
4. Determining Iteration Planning and Team Sync locations and timing.

1.7.17 [60 minutes] Define release objectives

After planning, the release engineer and other team of team stakeholders summarize the individual team objectives into a set of release objectives and use this to communicate externally and track progress toward the goals.

1.8 POST EVENT ACTIONS

1.8.1 Clean up the rooms used for planning (if applicable).

1.8.2 Enter the team release objectives and stories in Agile lifecycle management (ALM) tooling.

1.8.3 Review team and team of team events calendars.

1.8.4 Determine Iteration Planning and Team Sync locations and timing.

1.9 OUTPUTS

1.9.1 **Committed release objectives** – Each team creates a set of SMART (i.e., Specific, Measurable, Achievable, Relevant, and Time-Bound) objectives with the business value assigned by the Business Owners.

1.9.2 **Team of teams planning board** – Highlighting the new feature delivery dates, feature dependencies among teams, and relevant milestones.

RELEASE LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 2 - Release

2.0 RELEASE REFINEMENT

2.1 PURPOSE

The purpose of release refinement is for teams of teams to take some time during the release to plan for the future, although teams of teams should be primarily focused on the current release goals. Release refinement sessions allow teams of teams to begin building alignment and shared understanding about upcoming work, to start identifying dependencies, to inform backlog prioritization, and to begin hypothesizing how teams of teams may address future issues. Product Managers (PMs) typically facilitate release refinement for the teams of teams (however, release engineers are suitable facilitators, and teams of teams should do it with or without a formal facilitator on the scheduled date and time). Refinement activities often occur during the Release Sync, where Product Management and Product Owners identify new backlog items, revise others, and remove obsolete items. A well-maintained backlog is a prerequisite for successful future release planning and execution.

2.2 REVISION HISTORY

Author	Description	Initials

2.3 PERSONS AFFECTED

Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary).

2.4 POLICY

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

2.5 RESPONSIBILITIES

2.5.1 The roles and responsibilities include. Product Managers, Release Engineers, and Teams of Teams are responsible for Analyzing new feature requests, Reviewing current features, Identifying critical enablers, Applying structured acceptance testing criteria, Reprioritizing features based on new information, Preannouncing upcoming release plans and features, Deleting aging and no longer relevant items, and Adjusting business context and product management vision (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 release refinement that the team of teams agrees upon. Decide how often to refine the release backlog. Hold a backlog refinement event at least once per release, but you can adjust frequency and duration based on the team of team's needs.
- 2.6.3 Prepare the release backlog. Visit the release 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 features for the next release. Begin prepopulating the release backlog with features 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 of teams capacity allocation. Gather data on the team of team's capacity and load for the upcoming release based on changing schedules, priorities, and other unplanned personal, professional, and administrative priorities.

2.7 PROCEDURE

- 2.7.1 Analyze new feature requests

Unplanned features often emerge from external and internal stakeholders during the execution of a release. These must be considered, added to the current release if possible, and the scope of the release must be changed to maintain a sustainable pace. It's important to maintain release predictability (i.e., the ability to satisfy release objectives).

2.7.2 Review current features

Review and update release backlog item definitions, review and adjust acceptance criteria and benefit hypothesis, and split them as necessary to ensure release predictability and updated customer needs and assumptions.

2.7.3 Identify critical enablers

Identify the enablers required to support new features and capabilities. This is especially true of architectural runways, along with team of teams as well as cross team of teams platforms and shared services.

2.7.4 Apply structured acceptance testing criteria

Apply Behavior-Driven Development (BDD) techniques to help clarify features and capabilities or holding specification workshops.

2.7.5 Reprioritize features based on new information

Prioritize the backlogs using Weighted Shortest Job First (WSJF) in collaboration with Business Owners, System Architects, POs, and other stakeholders, such as solution engineers and release engineers.

2.7.6 Preannounce upcoming release plans and features

Brief Agile Teams and stakeholders about upcoming features and capabilities for Release Planning.

2.7.7 Delete aging and no longer relevant items

Proactively evaluate the present and future release backlogs, ensure work items are systematically evaluated and priorities, align them with present and future needs, and cull any features or acceptance criteria that are not aligned with the product management vision, goals, and business context.

2.7.8 Adjust business context and product management vision (as necessary)

Review the business context and product management vision and make any adjustments as necessary based on changing release objectives in the past, present, and future releases (for sharp alignment and solution acceptance).

2.8 POST EVENT ACTIONS

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

2.8.2 Communicate with other teams of 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 of teams release goals and objectives in the upcoming releases(s).

2.9.2 Shared understanding of how upcoming work may be solutioned.

2.9.3 Estimated features with acceptance criteria ready for the next release.

2.9.4 Enablers and spikes for further investigation.

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

RELEASE LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 2 - Release

3.0 RELEASE SYNC

3.1 PURPOSE

The purpose of the Release Sync is to help the team of teams to stay on track during a release (which is typically a calendar quarter or 90-day period consisting of up to five two-week iterations and a three-week innovation and planning iteration). It is typically facilitated by a release engineer and/or product manager once or twice a week. This is an opportunity to bring team of team leaders together with important representatives from the teams, focus on features and progress towards planning release objectives, and align individual agile teams on a team of teams. Attending product owners (POs) manage the release's scope, review progress, adjust priorities, and prepare for the following release. Agile coaches or scrummasters manage release execution, risks, dependencies, progress, and impediments. After the agreed-upon timebox with the whole group, time should be reserved for meet-afters on any topic that requires deeper discussion.

3.2 REVISION HISTORY

Author	Description	Initials

3.3 PERSONS AFFECTED

Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary).

3.4 POLICY

3.4.1 The policy of this organization is to ensure. Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Release Sync events (in order to manage the release's scope, adjust priorities, review the team of team's progress, track progress on a weekly basis, coordinate dependencies, resolve impediments, ensure the team of teams is operating at optimal velocity, and the team of teams on accomplishing the release goals or objectives, and prepare for the following release).

3.5 RESPONSIBILITIES

3.5.1 The roles and responsibilities include. Product Managers, Release Engineers, Teams of Teams are responsible for Reporting progress on features, Identifying any changes to scope (as needed), Reviewing release plans, Adjusting release plan and dependencies (as needed), Preparing for the next planning event, Reviewing process on release goals, and Facilitating meet-afters (as necessary).

3.6 INPUTS

- 3.6.1 **Select an optimal time.** Establish when your team of teams will hold release syncs throughout the release and schedule these event times. Make sure POs, PMs, scrum masters and team coaches, and important stakeholders know when these syncs will be held.
- 3.6.2 **Identify team representatives to participate.** Ensure that every team can be represented at release sync by a PO, scrum master or team coach, and/or team lead.
- 3.6.3 **Invite key subject matter experts.** Invite representatives from engineering and architecture to the meeting, as they have key input about how to move the work forward.
- 3.6.4 **Decide who will facilitate each release sync.** Facilitation may be a shared responsibility. Whomever is facilitating any part of an upcoming release sync should know in advance that they will be facilitating.
- 3.6.5 **Prepare a standard or tailored agenda.** Communicate the meeting topics to the attendees ahead of the next sync.
- 3.6.6 **Apply a standard agenda format.** Create an adjustable but re-usable release sync agenda template, and make sure you update it before the upcoming sync to facilitate an organized flow of discussion.

3.7 PROCEDURE

- 3.7.1 **Report progress on features**
Discuss features teams are working on and highlighting dependencies, blockers, or risks.
- 3.7.2 **Identify any changes to scope (as needed)**

Ask if product managers or product owners need to suggest scope adjustments to features.

3.7.3 Review release plans

Review release, integration, test, evaluation, and deployment plans.

3.7.4 Adjust release plan and dependencies (as needed)

Align all agile teams to the team of teams release planning board.

3.7.5 Prepare for the next planning event

Begin preparing release backlog for the next team of teams release planning event.

3.7.6 Review process on release goals

Reviewing progress towards team of teams-level release goals or objectives.

3.7.7 Facilitate meet-afters (as necessary)

After the agreed-upon timebox with the whole group, time should be reserved for meet-afters on any topic that requires deeper discussion.

3.8 POST EVENT ACTIONS

3.8.1 Release engineer follows up on risks and impediments.

3.8.2 Product manager updates release goals, if required.

3.8.3 Communicate possible impacts to dependent teams or release objectives.

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

3.9 OUTPUTS

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

3.9.2 Surfaced risks/impediments that are blocking the team of teams.

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

RELEASE LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 2 - Release

4.0 SYSTEM DEMO

4.1 PURPOSE

The purpose of the System Demo is to provide stakeholders with an integrated view of new features for the most recent iteration delivered by all the teams on the team of teams. Each demo provides an objective measure of progress and the opportunity to give feedback. A system demo is a critical event. It's the method for assessing the Solution's current state and gathering immediate, team of team-level feedback from the people doing the work and critical feedback from Business Owners, sponsors, stakeholders, and Customers. The demo is the one objective measure of value, velocity, and progress of the fully integrated work across all the teams. Planning for and presenting a useful system demo requires work and preparation by the teams. But it's the only way to get the fast feedback needed to build the right solution.

4.2 REVISION HISTORY

Author	Description	Initials

4.3 PERSONS AFFECTED

Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary).

4.4 POLICY

4.4.1 The policy of this organization is to ensure. Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in System Demo events (in order to demonstrate measurable progress towards release goals, solicit stakeholder feedback, and influence future course corrections, release backlog refinement, and release plans).

4.5 RESPONSIBILITIES

4.5.1 The roles and responsibilities include. Product Owners, Scrum Masters, and Agile Teams are responsible for Describe features, Demoing each new feature in an end-to-end use case, Identifying Risks, Facilitating questions and answer session, and Concluding system demo.

4.6 INPUTS

4.6.1 Develop a standard system demo agenda. Create an adjustable but re-usable system demo agenda template, and make sure you update it before the system demo to create an organized flow of demos.

4.6.2 Identify features to be demoed in advance. Determine which features and capabilities can be shown at the system demo at the beginning of the iteration; don't wait until the end of the iteration when teams are busy finishing up work.

4.6.3 Prepare system demo facilities. Identify who will facilitate or lead the upcoming system demo in advance so they can prepare to lead the meeting and do their own technology check.

4.7 PROCEDURE

4.7.1 [5-10 minutes] Describe business context

Review the business context and the release objectives for the current team of teams release.

4.7.2 [5 minutes] Describe features

Describe each new feature briefly before demoing to the entire of team of teams (including stakeholders).

4.7.3 [20-30 minutes] Demo each new feature in an end-to-end use case

Create a per-demo timebox of 5–10 minutes per feature being demonstrated.

4.7.4 [10 minutes] Identify Risks

Identify current risks and impediments.

4.7.5 [10 minutes] Facilitate questions and answer session

Open discussion of questions and feedback.

4.7.6 [5 minutes] Conclude system demo

Wrap up by summarizing progress, feedback, and action items.

4.8 POST EVENT ACTIONS

- 4.8.1 Update the release backlog based on feedback from the system demo.
- 4.8.2 Reflect on the unfinished features 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 inspect & adapt and consider how future releases can be better planned and executed.
- 4.8.4 Identify opportunities to show integrated work at the solution or enterprise demo.
- 4.8.5 Move the work not completed to the release backlog for consideration in future releases.

4.9 OUTPUTS

- 4.9.1 Planned release goals and work items.
- 4.9.2 Accomplished release goals and work items.
- 4.9.3 Impediments or dependencies preventing completion of release goals and work items.
- 4.9.4 Goals, risks, dependencies, impediments, changes to the release backlog, new priorities, and work items for next release.
- 4.9.5 Release metrics such as historical, planned, added, deleted, and actual business value and velocity. This may include metrics such as team morale, quality, customer satisfaction, etc.

RELEASE LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number	
Revision	
Approved by	
Date	

AGILE-DEV 1.0

Level 2 - Release

5.0 INSPECT & ADAPT

5.1 PURPOSE

The purpose of Inspect & Adapt is to demonstrate and evaluate the current state of the solution. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop. The Agile Manifesto recommends reflecting on how to become more effective. Then, tuning and adjusting behavior. Relentless improvement is a core value and dimension of a continuous learning culture. Continuous improvement is an ongoing process throughout a release, especially as the need arises. However, time should be explicitly set aside at the end of a release for an Inspect & Adapt event.

5.2 REVISION HISTORY

Author	Description	Initials

5.3 PERSONS AFFECTED

Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary).

5.4 POLICY

5.4.1 **The policy of this organization is to ensure.** Product Managers, Release Engineers, Teams of Teams, and Stakeholders (as necessary) shall facilitate, execute, and participate in Inspect & Adapt events (in order to identify what's working well, what's not working well, and how to improve the next release's performance to achieve release goals efficiently and effectively, optimize teams of team's performance, and maximize resulting value).

5.5 RESPONSIBILITIES

5.5.1 **The roles and responsibilities include.** Product Managers, Release Engineers, and Teams of Teams are responsible for Facilitating system demo, Evaluating quantitative and qualitative release performance, Facilitating retrospective, and Facilitating problem-solving workshop.

5.6 INPUTS

5.6.1 **Schedule the I&A event far in advance.** This is an important event for the entire team of teams, and it has a significant timebox. Scheduling well in advance ensures that everyone can participate reliably.

5.6.2 **Decide if your I&A will be held in person or as a remote event.** Plan how you will set up the room and/or remote meeting platform. If the I&A is across multiple office locations, consider having an onsite facilitator at each location.

5.6.3 **Prepare technology needs in advance.** Ensure you have planned technology support to present slides, demos, videos, or other artifacts to the team of teams.

5.6.4 **Identify technology support personnel.** You may need a designated support person to ensure that any screens or sound system are working as needed.

5.6.5 **Ensure business owners are present.** Work with the business owners to schedule meetings with each Agile team so they can score the actual business value achieved by each team. This must be completed for each team on the team of teams before the I&A in order to roll up the team of team's business value in the quantitative and qualitative evaluation portion of the event.

5.6.6 **Prepare a live operational system demo.** Work with the teams, product management, system and solution architects, engineering, and other key contributors to plan and prepare the release system demo.

5.6.7 **Identify support facilitators.** Identify and designate facilitators and scrum masters or team coaches for each team or small group who will complete the retrospective and problem-solving workshop.

5.6.8 **Identify a post-action plan.** Plan for how the team of teams will take improvement items into the backlog, how they will identify owners and collaborators to work on them and share progress or changes with the team of teams as these items are considered.

5.7 PROCEDURE

5.7.1 **[5 minutes]** Welcome all participants

Remind everyone of the purpose of this event and give them a brief overview of the agenda.

5.7.2 [60 minutes] Facilitate system demo

The I&A starts with a system demo to present the current state of the solution. This demo is intended to show all the features that the team of teams has developed in the release. The system demo also tends to be more formal than the demos that happen after each iteration, and some extra preparation is often required. Although there are some differences between this and other system demos, it should be timeboxed to an hour or less to keep everyone engaged.

5.7.3 [20-45 minutes] Evaluate quantitative and qualitative release performance

Next is a collective review of any metrics the team of teams has agreed to track, and a discussion of the data and trends. The release engineer and/or solution engineer are typically responsible for gathering and presenting this information to the team of teams. The metrics that teams of teams decide to track vary widely. One strongly recommended is the team of team's predictability measure. This is gathered by rolling up each agile team's planned vs. actual business value. Reliable teams of teams should operate in the 80-100 percent range, as this allows the business and its stakeholders to plan most effectively.

5.7.4 [20-30 minutes] Facilitate retrospective

Teams of teams should conduct a brief retrospective (30 minutes or less) to reflect on and surface the issues they would like to address during the problem-solving workshop. There is no one way to do this; many different Agile retrospective formats can be used.

5.7.5 [60+ minutes] Facilitate problem-solving workshop

The problem-solving workshop helps address systemic issues using the structured root cause analysis format. This consists of agreeing on the problem to solve, performing a root cause analysis, identifying the biggest root cause, restating the new problem, brainstorming solutions, and then generating improvement items to add to the backlog.

5.8 POST EVENT ACTIONS

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

5.8.2 Find a way to track improvement items the team of teams 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 of teams 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 solution or enterprise level, work with the product manager (PM) to share these with the solution, portfolio, or enterprise engineer.

5.9 OUTPUTS

5.9.1 The retrospective facilitator and the team of team's product manager (PM) should gain one or two improvement epics, features, or stories to add to the release backlog.

5.9.2 Follow up with the product manager (PM) to make sure they have what they need to add these the epics, features, or stories for release backlog refinement.

5.9.3 Offer your team of teams 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 of teams.