

# TEAM LEVEL AGILE DEVELOPMENT FRAMEWORK — VERSION 1.0

Number

Revision

Approved by

Date

AGILE-DEV 1.0

Level 1 - Team

1.0 PRODUCT OWNER (PO) ROLES AND RESPONSIBILITIES

## 1.1 PURPOSE

The purpose of this policy and procedure is to establish a minimum set of roles and responsibilities for Product Owners (POs) within the context of a larger agile team of teams (i.e., a product or project consisting of two or more Scrum teams). It establishes the PO roles and responsibilities for release planning preparation, release planning, release execution, release refinement, and release wrapup activities. A PO is a technical, Scrum development team facing role responsible for release and Sprint backlogs for one or more Scrum development teams. POs are responsible for interfacing to a chief PO also known as an Agile Product Manager (APM). The APM often reports to a higher-level traditional product manager responsible for marketing and direct customer and client interaction. APMs are responsible for conducting Lean UX activities to determine the scope of a minimum viable product (MVP) which is often broken down into major subsystems. POs typically have responsibility for one or more subsystems but not the whole system MVP. Therefore, POs are responsible for determining the scope of an MVP's subsystem(s) to which they've been allocated. PO responsibilities include facilitating subsystem architecture and design activities, decomposing subsystem designs into work breakdown structures, and formulating associated project network diagrams. From these basic artifacts, POs are responsible for identifying features, story maps, draft release plans, and final release plans. Once final release plans are formed, POs are responsible for executing the release plan by participating in major Scrum team events such as Sprint Planning, Team Syncs, Sprint Reviews, Retrospectives, and Backlog Refinement events. Scrummasters are responsible for establishing and facilitating these events, process integrity, resolving impediments, and overall process and team health, productivity, effectiveness, efficiency, and performance. POs determine the scope of a Scrum team's effort generally in the form of Product Backlogs, Sprint Goals, Features, User Stories, and Acceptance Criteria. POs monitor the satisfaction of Sprint Goals, optimizing Scrum team capacity and load, the flow of work, and the satisfaction of release goals. POs also manage the reception of ad hoc, unplanned work, are responsible for triaging and scoping it, and determining whether new work should be performed now, next, or later. POs are also responsible for managing product dependencies, impediments, and risks. POs also interface to other POs, PMs, and the team of teams.

## 1.2 REVISION HISTORY

Author	Description	Initials

## 1.3 PERSONS AFFECTED

Product Owners (POs), Scrummasters, Scrum Team Members, Product Managers, Release Train Engineers, other Key Stakeholders, etc.

## 1.4 POLICY

The policy of this organization is to ensure that:

1.4.1 Product Owners (POs) perform Release Planning Preparation, Release Planning, Release Execution, Release Refinement, and Release Wrapup and Closure activities.

## 1.5 RESPONSIBILITIES

1.5.1 Product Owners (POs) are Responsible for Release Planning Preparation activities, such as Contribute to Release Roadmap, Translate Roadmap into Features, Facilitate Feature and Story Mapping, and Prepare Release Network Models.

1.5.2 Product Owners (POs) are Responsible for Release Planning, such as Facilitate Release Preplanning, Facilitate Release Planning, and Populate Release Backlogs.

1.5.3 Product Owners (POs) are Responsible for Release Execution activities, such as Participate in Sprint Planning, Participate in Team Syncs, Participate in Sprint Reviews, Participate in Retrospectives, Facilitate Team Backlog Refinement, and Prepare and Deliver Sprint Accomplishments.

1.5.4 Product Owners (POs) are Responsible for Release Refinement activities, such as Participate in Product Owner Syncs, and Participates in Inter-team Backlog Refinement, Manage Goals, Resources, Dependencies, and Risks, Triage Ad Hoc Requests for New Features and Stories, and Facilitate Ad Hoc Feature and Story Mapping.

1.5.5 Product Owners (POs) are Responsible for Release Wrapup and Closure, such as Prepare and Deliver Release Accomplishments, Measure Release and Sprint Performance, and Participate in Release Retrospectives.

## 1.6 INPUTS

- 1.6.1 **Product Vision.** A short, but impactful vision statement for the product, generally in the form of a short statement, simple narrative, or elevator speech.
- 1.6.2 **Product Canvas.** A lean canvas or business model with a short description of the product problem, solution, unique value proposition, advantage, customer segments, key metrics, channels, cost structure, and revenue streams.
- 1.6.3 **Product Value Stream.** A value stream map for the major cradle to grave delivery steps, phases, or stages for operating the product to receive the desired customer output or outcome and the features or subsystems required to enable it.
- 1.6.4 **Product Architecture.** An overall architecture and design layout of the minimum viable product (MVP), its inputs and outputs, major subsystems, interfaces, and performance constraints.
- 1.6.5 **Product Roadmap.** A high level incremental and iterative roadmap illustrating the major features of the MVP and the timelines or order in which the features are to be developed and delivered.
- 1.6.6 **Product Features.** A list of high priority, value adding features for each PO, Scrum team or group of POs and Scrum teams in which to collaboratively develop.
- 1.6.7 **Assumptions and Constraints.** The key assumptions and constraints of the features including what is in and out of scope, performance, and other technical performance characteristics.

## 1.7 PROCEDURE

### RELEASE PLANNING PREPARATION

#### 1.7.1 Contribute to Release Roadmap

##### 1.7.1.1 Identify Product Milestones and Goals

Identify current list of major program milestones, inchstones, integration points, test activities, deliveries, and other special events.

##### 1.7.1.2 Identify Recurring and Future Features

Identify any recurring work such as system and server maintenance, IT support, customer support, operations, maintenance, and other necessary technical debt.

##### 1.7.1.3 Align Features with Product Milestones and Goals

Ensure major features for planning, recurring, and even unplanned events are noted in roadmap and aligned with milestones and goals.

##### 1.7.1.4 Develop Roadmap or Timeline for Features

Create or contribute to a roadmap illustrating program milestones, goals, special events, release dates, features, etc. in ALM or similar medium.

##### 1.7.1.5 Describe Features, Dependencies, and Risks

Ensure features in roadmap contain basic descriptions, benefit value statements, dependencies, risks, parent epic identification numbers, and other important data.

##### 1.7.1.6 Review and Verify Roadmap with Stakeholders

Solicit feedback from product owners, product managers, technical staff, teams, and agile product management office (APMO) staff and members.

##### 1.7.1.7 Communicate Next Release Features to Scrum Teams

Distribute, describe, and publish roadmap to managerial, administrative, and technical leaders (and keep it up to date as new information evolves).

#### 1.7.2 Translate Roadmap into Features

##### 1.7.2.1 Identify List of Recurring and New Features

Gather and review current roadmap with features, emerging priorities opportunities, needs, and events, and cull a list of current release features.

##### 1.7.2.2 Clarify Feature Titles, Descriptions, Dependencies, & Risks

Clarify and update the titles, descriptions, scope, and constraints of planned, unplanned, and emerging features for current release.

##### 1.7.2.3 Add Preliminary Feature Acceptance Criteria (if Available)

Begin to identify the scope, bounds, constraints, and stories necessary to complete each feature in roadmap or updated release priorities.

**1.7.2.4 Record Recurring and Future Features in Backlog**

Record currently approved features list in ALM workflow tools with titles, descriptions, dependencies, risks, and preliminary acceptance criteria.

**1.7.2.5 Ensure Recurring and Future Features are Tagged**

If using a dynamic ALM workflow tool, ensure to annotate each feature entry with tags for release, milestones, deliveries, teams, parent epics, etc.

**1.7.2.6 Verify Recurring and Future Features Aligned with Roadmap**

Ensure static roadmaps are aligned with features in ALMs if separate ALM tools are used (although advanced ALM tools integrate roadmaps and features together).

**1.7.2.7 Maintain Alignment Between Roadmap and Backlog Features**

Ensure roadmaps and ALM feature backlogs are aligned as new data, information, priorities, and special events occur impacting roadmap artifacts.

**1.7.3 Facilitate Feature and Story Mapping****1.7.3.1 Identify List of Features for Current Release**

Begin to cull and finalize list of features for current release from roadmaps, stakeholder inputs, and other emerging events impacting current release.

**1.7.3.2 Prepare Draft Story Maps for Each Team's Features**

Begin to identify list of stories and critical activities for each major, known feature, remove unnecessary stories, and identify ad hoc work too.

**1.7.3.3 Review Story Maps with Subject Matter Experts (SMEs)**

Begin peer reviewing feature and story maps with key technical and non-technical stakeholders to confirm assumptions and update story maps and lists.

**1.7.3.4 Review Story Maps with Each Scrum Team**

Continue peer reviewing feature and story maps with Scrum teams and all technical developers to solicit additional feedback and socialize scope of release activities.

**1.7.3.5 Review Story Maps with Product Managers and Stakeholders**

Peer review feature and story maps with product managers, system architects, and other key stakeholders so they can incorporate stories into plans.

**1.7.3.6 Distribute and Communicate Story Maps to Stakeholders**

Assemble and publish early lists of high priority features and story maps to key stakeholders for review, consideration, improvement, and refactoring.

**1.7.3.7 Update Features in Backlogs with Story Map Data**

Update feature backlogs in ALM tools with new information if stable and appropriate (and update acceptance criteria to reflect scope of approved features).

**1.7.4 Prepare Release Network Models****1.7.4.1 Identify List of Features for Current Release**

Ensure network diagramming subject matter experts (SMEs) have updated list of high priority features for current release for planning purposes.

**1.7.4.2 Identify List of Stories for Current Features**

Ensure network diagramming subject matter experts (SMEs) have updated list of high priority stories and activities for current release for planning purposes.

**1.7.4.3 Organize Features and Stories into Work Breakdown Structure**

Assist subject matter experts (SMEs) with organizing high priority features and stories for current release into integrated work breakdown structures.

**1.7.4.4 Identify Timelines and Dependencies Between Features and Stories**

Assist subject matter experts (SMEs) with identifying timelines, resources, dependencies, milestones, and other constraints for features and stories.

**1.7.4.5 Organize Data into a Project Network Diagram**

Assist subject matter experts (SMEs) with organizing work breakdown structures, features, stories, and constraints into project network diagrams for current release.

**1.7.4.6 Review and Verify Project Network Diagram with Stakeholders**

Review draft project network diagrams, socialize with key technical and non-technical stakeholders, and solicit feedback and corrections for release.

**1.7.4.7 Distribute and Communicate Project Network Diagrams to Teams**

Receive and further promulgate approved project network diagrams to key stakeholders to inform release preplanning events, artifacts, and activities.

## RELEASE PLANNING

### 1.7.5 Facilitate Release Preplanning

#### 1.7.5.1 Identify Vetted Project Network Diagrams

Ensure subject matter experts (SMEs) have created, updated, and vetted a reasonably approved project network diagram for the features and stories of the current release.

#### 1.7.5.2 Prepare Release Plans for Scrum Teams

Analyze project network diagram and begin decomposing it into features, stories, and prerelease plans for the current release (as it represents an approved release plan).

#### 1.7.5.3 List Features, Dependencies, and Risks

Clarify features, stories, dependencies, and risks for each major feature or story in draft release plan (as this represents a first cut at an approved release plan).

#### 1.7.5.4 Identify Vetted Stories for Each Feature

Elaborate stories for each feature from project network diagram, verify with subject matter experts (SMEs), close gaps, and begin finalizing feature acceptance criteria.

#### 1.7.5.5 Determine Sprint for Each Story

Identify and preassign stories into Sprints based on dates, milestones, and integration points in project network diagram (and split larger stories into multiple Sprints).

#### 1.7.5.6 Review Goals, Features, and Stories for Each Team

Review draft release plans with Product Owners (POs), Scrummasters, key technical personnel, and subject matter experts (SMEs) for greater accuracy and precision.

#### 1.7.5.7 Distribute Draft Release Plans to Stakeholders for Review

Distribute and communicate draft release plans to key stakeholders and Scrum teams, hold private peer review sessions, and present in Scrum team meetings.

### 1.7.6 Facilitate Release Planning

#### 1.7.6.1 Determine Release Capacity for Each Scrum Team

Elicit capacity for each Scrum team and developer for the release and each Sprint based on holidays, leave, and other special events throughout current release.

#### 1.7.6.2 Organize Sprints and Assign Stories to Each One

Verify and socialize Sprints in which each story is assigned based on project network diagrams in feature and story maps (as well as Sprint backlogs and ALM tools).

#### 1.7.6.3 Draft User Story Titles and Associated Features

Elicit clarification on draft user story titles, descriptions in user story format, and acceptance criteria (which should be drafted in Sprint plans and ALM tools).

#### 1.7.6.4 Estimate or Seed Story Points Based on Historical Data

Facilitate user story sizing from Scrum teams using Planning Poker (which are directly informed by release network diagrams, durations, and historical data).

#### 1.7.6.5 Split Stories Across Multiple Sprints for Large Stories

Split and create multiple stories for larger user stories that span multiple Sprints (which may be preferable to keeping stories open for multiple Sprints).

#### 1.7.6.6 Identify Goals, Resources, Dependencies, and Risks

Draft, vet, and finalize release and Sprint goals, resources, dependencies, and risks with key stakeholders, Product Owners (POs), and Scrum team technical members.

#### 1.7.6.7 Review Release and Sprint Plans with Stakeholders for Approval

Present release plans to major stakeholders as well as members of all release Scrum teams for review, feedback, and improvements before release begins.

### 1.7.7 Populate Release Backlogs

#### 1.7.7.1 Identify List of Features and Stories from Release Planning

Identify current list of approved and vetted features and user stories from release planning and technical and non-technical stakeholder reviews.

#### 1.7.7.2 Prepare Release Plans in Agile Lifecycle Management (ALM) Tool(s)

Configure and format ALM tools for current release and Sprint with correct number of Sprints, release and Sprint

dates, and Sprint goals.

#### 1.7.7.3 **Create and Align Features and Stories in ALM Tool**

Create all approved and vetted features and user stories in ALM tools and align them with the correct release and Sprints according to date and Sprint goal.

#### 1.7.7.4 **Ensure Stories Have Acceptance Criteria, Sprints, Team, and Tags**

Ensure all approved and vetted features and user stories have acceptance criteria, release tags, Sprint tags, team tags, and other special tags.

#### 1.7.7.5 **Ensure Features and Stories are Sized and Accurate**

Ensure all approved and vetted user stories have story point sizes, are properly entered into ALM tools, and match a capacity and load estimates in other ALM tools.

#### 1.7.7.6 **Review Feature and Story Descriptions with Scrum Teams**

Distribute, communicate, and review release and Sprint plans in ALM tools with subject matter experts (SMEs), Product Owners (POs), and Scrum team developers.

#### 1.7.7.7 **Review Feature and Story Descriptions with Stakeholders**

Distribute, communicate, and review release and Sprint plans in ALM tools with wider stakeholder community for tracking and performance measurement purposes.

### RELEASE EXECUTION

#### 1.7.8 Participate in Sprint Planning

##### 1.7.8.1 **Participate in Recurring Meetings**

Participate in recurring Sprint planning meetings on the first day of each Sprint to clarify scope, selection and sizing of stories, and a feasible high-value Sprint plan is formulated.

##### 1.7.8.2 **Prepare Draft Sprint Plans**

Collaborate with technical leaders, chief programmers, and technical subject matter experts (SMEs) to formulate a draft Sprint plan, stories, tasks, story points, acceptance criteria, etc.

##### 1.7.8.3 **Review Release and Sprint Goals**

Review and confirm planned release and Sprint goals with Scrum teams to help determine the scope, stories, tasks, and previous commitments made during major release planning events.

##### 1.7.8.4 **Verify Availability and Capacity**

Identify availability, capacity, and load of technical Scrum team members and developers to select and scope appropriately sized user stories, tasks, acceptance criteria, and goals.

##### 1.7.8.5 **Select and Size User Stories**

Select high priority stories from release and Sprint plans, elicit Scrum team input and story point sizing, and determine which stories to accept, assign, and allocate to current Sprint.

##### 1.7.8.6 **Clarify Acceptance Criteria**

Clarify user story descriptions, acceptance criteria, and scope to help Scrum team members understand commitments, availability, and alignment with goals (which may require follow on meetings).

##### 1.7.8.7 **Populate and Verify Sprint Backlogs**

Populate and verify selected user stories from planned release and Sprint plans as well as current Sprint planning event, acceptance criteria, story point sizes, and Sprint goals.

#### 1.7.9 Participate in Team Syncs

##### 1.7.9.1 **Participate in Recurring Meetings**

Participate in recurring Team Syncs throughout the Sprint duration to understand and clarify the work Scrum teams are performing, progress towards goals, emerging impediments, etc.

##### 1.7.9.2 **Review and Analyze Sprint Plans**

Review and analyze the current Sprint plan as a Scrum team, identify the stories each person is working on, ensure user story progress, identify impediments, and identify emerging work.

##### 1.7.9.3 **Participate in Round Robin Reporting**

Participate in round robin reporting to listen to the progress of work on user stories, identify technical issues, clarify acceptance criteria, and identify emerging impediments.

##### 1.7.9.4 **Identify New Impediments, Dependencies, and Risks**

Assist Scrummasters in identifying new and emerging impediments, dependencies, and risks which may hinder the accomplishment of release and Sprint goals and stories.

##### 1.7.9.5 **Update Status of Sprint Plan**

Correct and update the status of Sprint plans to annotate completed tasks, acceptance criteria, and stories (as well as emerging impediments, blockages, and work deletion or additions).

#### **1.7.9.6 Add or Remove User Stories (if necessary)**

Add, split, or remove stories as necessary to align them with Sprint goals, ensure Sprint goals are achievable, and match the capacity to demand and load for optimal velocity.

#### **1.7.9.7 Co-Manage Dependencies and Risks**

Assist Scrummaster in resolving critical dependencies and risks associated with successfully achieving release and Sprint goals (and make needed release, Sprint, and backlog adjustments).

### **1.7.10 Participate in Sprint Reviews**

#### **1.7.10.1 Participate in Recurring Meetings**

Participate in recurring Sprint review meetings on the last day of each Sprint to review completed stories, view operational demos, evaluate Sprint accomplishments, and make adjustments.

#### **1.7.10.2 Verify Team is Ready for Demos**

Assist Scrummasters in determining which stories will be demonstrated, ensure Scrum team members can do this quickly and succinctly, and optimize the performance and effectiveness of Sprints.

#### **1.7.10.3 Review and Analyze Sprint Plans**

Review and analyze Sprint plan as a team, identify which stories are complete, identify which will be demonstrated and completed, and what work needs to be split, backlogged, or continued.

#### **1.7.10.4 Participate in Round Robin Demos**

Review each user story one-by-one, have the Scrum team member speak to it, demonstrate a static or dynamic result, determine whether to accept it, and identify whether to pivot or persevere.

#### **1.7.10.5 Accept and Close User Stories**

Analyze each user story, verify acceptance criteria, consider user story demonstration results, accept and close user stories as necessary, and split, backlog, or continue unfinished work.

#### **1.7.10.6 Split or Move User Stories**

Assist the Scrummaster in determining if acceptance criteria can be split or rejected, whether to accept stories as is, split or backlog them if necessary, or create smaller follow-on stories.

#### **1.7.10.7 Close Sprint and Records Metrics**

Assist Scrummaster in closing Sprints in ALM systems, recording the goals that have been achieved, recording performance metrics, and preparing release and Sprint backlogs for next Sprint.

### **1.7.11 Participate in Retrospectives**

#### **1.7.11.1 Participate in Recurring Meetings**

Participate in recurring Sprint retrospective meetings on the last day of each Sprint to analyze, identify, manage, and resolve high priority process and product improvement opportunities.

#### **1.7.11.2 Review and Analyze Retrospective Boards**

Access retrospective boards, tools, and other mediums in order to review and analyze strengths, weaknesses, and process and product opportunities identified by Scrum team members.

#### **1.7.11.3 Analyze Strengths, Weaknesses, and Improvements**

Visually analyze inputs to retrospective boards to understand strengths, weaknesses, and possible high priority, value adding process and product improvement opportunities.

#### **1.7.11.4 Help Prioritize and Identify Improvements**

Observe and analyze Scrum team technical members as they vote on process and product improvement opportunities in order to log, manage, and resolve them for optimizing release or Sprint performance.

#### **1.7.11.5 Communicate Product Improvements to Stakeholders**

Take ownership of high-priority, value adding product improvement opportunities, communicate them to stakeholders, and help resolve and close them in a timely manner (if possible).

#### **1.7.11.6 Facilitate Product Improvement Initiatives**

Establish one or more technical exchanges with product managers and product owners to resolve high priority, value adding product improvement opportunities (and communicate to Scrum team).

#### **1.7.11.7 Help Update Status of Retrospective Boards**

Update the status of initiatives and attempts to resolve and close high-priority, value adding product improvements (and communicate updated retrospective issues to stakeholders).



### 1.7.12 Facilitate Team Backlog Refinement

#### 1.7.12.1 Participate in Recurring Meetings

Participate in recurring Team backlog refinement meetings to review progress of current Sprint toward goals and needs and opportunities to adjust and adapt Sprint and product backlogs.

#### 1.7.12.2 Review and Analyze Sprint Plans

Review and analyze progress of current Sprint to determine whether user stories are completed or blocked and need to be accepted or moved to the next Sprint, release, or product backlog.

#### 1.7.12.3 Review and Analyze Product Backlogs

Review and analyze release or product backlogs to ensure highest priority user stories are at the top, they are scoped and sized, have acceptance criteria, and readiness to begin user stories.

#### 1.7.12.4 Review and Analyze Test Defects

Review and analyze test defect backlogs to ensure highest priority test defects are at the top, they are scoped and sized, have acceptance criteria, and readiness to resolve test defects.

#### 1.7.12.5 Review and Analyze CVEs and IAVAs

Review and analyze CVE and IAVA backlogs to ensure highest priority CVEs and IAVAs are at the top, they are scoped and sized, have acceptance criteria, and readiness to resolve CVEs and IAVAs.

#### 1.7.12.6 Review and Analyze Technical Debt

Review and analyze technical debt backlogs to ensure highest priority technical debt is the top, it is scoped and sized, has acceptance criteria, and tech debt is ready to be accepted.

#### 1.7.12.7 Adjust Product and Sprint Backlogs

Elicit capacity and load of Scrum team members, facilitate selection of new stories, test defects, CVEs, IAVAs, and technical debt, and add new stories to release and Sprint backlogs.

### 1.7.13 Prepare and Deliver Sprint Accomplishments

#### 1.7.13.1 Review and Analyze Sprint Plans

Review and analyze current Sprint plans, identify user stories that have been completed or will be completed soon, get up-to-date information from Scrum team, and identify completed work.

#### 1.7.13.2 Gathers Data from Scrum Team

Gather up-to-date real-time information and data on any user stories, tasks, ad hoc, out-of-band, significant, major, or high-priority value adding work that has been completed (or will be completed in current Sprint).

#### 1.7.13.3 Identify Sprint Goals

Review and record current Sprint goals to understand Scrum team commitments, determine what user stories are in play, which have been completed, and whether Sprint goals have been satisfied to-date.

#### 1.7.13.4 Draft Sprint Accomplishments

Cull all information and data regarding completed Sprint goals and user stories, and other out-of-band accomplishments, and prepare a draft of high-impact Scrum team Sprint accomplishments.

#### 1.7.13.5 Identify Blockers

Identify any high-impact impediments or blockers that are hindering the current release or Sprint, preventing accomplishment of release or Sprint goals, and recommended actions to resolve impediments.

#### 1.7.13.6 Identify Upcoming Sprint Goals

Review release and Sprint plans, identify planned Sprint goals, review current (adjusted) Sprint goals from Sprint reviews and refinements, and identify the team goals for the next Sprint.

#### 1.7.13.7 Verify, Distribute, and Deliver Report

Verify Sprint accomplishments, prepare and distribute final reports, identify a reporter for Sprint accomplishments, and participate in periodic recurring team-of-teams readouts.

## RELEASE REFINEMENT

### 1.7.14 Participate in Product Owner Syncs

#### 1.7.14.1 Participate in Recurring Meetings

Participate in recurring Product owner syncs to identify product accomplishments, impediments, risks, dependencies, and emerging needs (to ensure release and Sprint objectives are on-track).

#### 1.7.14.2 Prepare Accomplishments and Status

Review and analyze progress of current Sprint goals and user stories to determine if release objectives and features are on track (or at risk due to dependencies, impediments, and risks).

#### 1.7.14.3 Prepare Dependencies, Impediments, and Risks

Review and identify current, emerging, and future dependencies, impediments, and risks (which may impact release objectives and features for team and overall team-of-teams).

#### **1.7.14.4 Participate in Round Robin Reporting**

Participate in round robin reporting to communicate status of release objectives, features, dependencies, impediments, and risks (so that product manager may work to resolve flow issues).

#### **1.7.14.5 Identify New Dependencies and Risks**

Gather new and emerging dependencies and risks from other teams in the team-of-teams that may impact release objectives and features (to other teams as well as your own team for resolution).

#### **1.7.14.6 Manage Dependency and Risk Resolution**

Proactively work to clarify, document, and resolve dependencies and risks to your team as well as other teams in the team-of-teams (using informal and formal issue resolution forums).

#### **1.7.14.7 Communicate Dependency and Risk Status**

Report status of current and emerging dependencies and risks to product managers or higher levels if necessary to verify and remediate impact of dependencies and risks (if necessary).

### **1.7.15 Participate in Inter-team Backlog Refinement**

#### **1.7.15.1 Participate in Recurring Meetings**

Participate in recurring Inter-team backlog refinement meetings to review progress of current Sprint toward goals and needs and opportunities to adjust and adapt Sprint and product backlogs.

#### **1.7.15.2 Review and Analyze Release Plans**

Review and analyze progress of current release to determine whether user stories are completed or blocked and need to be accepted or moved to the next Sprint, release, or product backlog.

#### **1.7.15.3 Review and Analyze Product Backlogs**

Review and analyze release or product backlogs to ensure highest priority user stories are at the top, they are scoped and sized, have acceptance criteria, and readiness to begin user stories.

#### **1.7.15.4 Review and Analyze Test Defects**

Review and analyze test defect backlogs to ensure highest priority test defects are at the top, they are scoped and sized, have acceptance criteria, and readiness to resolve test defects.

#### **1.7.15.5 Review and Analyze CVEs and IAVAs**

Review and analyze CVE and IAVA backlogs to ensure highest priority CVEs and IAVAs are at the top, they are scoped and sized, have acceptance criteria, and readiness to resolve CVEs and IAVAs.

#### **1.7.15.6 Review and Analyze Technical Debt**

Review and analyze technical debt backlogs to ensure highest priority technical debt is the top, it is scoped and sized, has acceptance criteria, and tech debt is ready to be accepted.

#### **1.7.15.7 Adjust Release and Product Backlogs**

Elicit capacity and load of Scrum team members, facilitate selection of new stories, test defects, CVEs, IAVAs, and technical debt, and add new stories to release and Sprint backlogs (as necessary).

### **1.7.16 Manage Goals, Resources, Dependencies, and Risks**

#### **1.7.16.1 Review Release and Sprint Plans on a Daily Basis**

Review and analyze release and Sprint plans on a daily basis to check the status of release and Sprint goals, features, user stories, dependencies, impediments, and risks.

#### **1.7.16.2 Ensure Release and Sprint Plans are On-Track**

Verify that release and Sprint plans are on track so that planned release and Sprint goals are achieved, features and user stories are completed, and performance goals are achieved.

#### **1.7.16.3 Manage Dependencies, Impediments, & Risks**

Manage known or emerging dependencies, impediments, and risks to ensure release and Sprint goals are achieved, features and user stories are completed, and performance is achieved.

#### **1.7.16.4 Ensure Stakeholders Understand Impediments and Risks**

Communicate known or emerging risks to technical Scrum team members or external teams to ensure release and Sprint goals, features, stories, and performance are achieved.

#### **1.7.16.5 Manage Stakeholder Expectations and Quality of Service**

Proactively seek to resolve dependencies, impediments, and risks to ensure other teams achieve release and Sprint goals, features, stories, and performance measures.

#### **1.7.16.6 Optimize Release and Sprint Plan Goals and Objectives**

Optimize load and capacity of Scrum teams for planned and emerging out-of-band work to ensure planning and



emerging release and Sprint goals and performance are achieved.

#### **1.7.16.7 Communicate Dependencies, Impediments, & Risks**

Record, communicate, and escalate the status of dependencies, impediments, and risks so that key stakeholders may help you accomplish release goals, objectives, and performance.

### 1.7.17 Triage Ad Hoc Requests for New Features and Stories

#### **1.7.17.1 Receive and Identify Ad Hoc Requests**

Establish governance, communication, tools, medium, and service levels for other teams to identify ad hoc requests, dependencies, and impediments, so that the greater team-of-teams can dynamically adapt to performance goals.

#### **1.7.17.2 Review them for Priority, Severity, and Date Needed**

Review all incoming ad hoc requests in timely manner to identify priority, severity, date needed, and level of service the requesting team may expect to receive from teams.

#### **1.7.17.3 Determine Size, Complexity, Feasibility, and Need**

Determine the size, complexity, feasibility, and need of ad hoc requests, forward to subject matter experts (SMEs) for further analysis, and determine capacity to deliver need.

#### **1.7.17.4 Add to Release or Sprint Backlog (if necessary)**

If the priority is high, the size and complexity are manageable, and the capacity and load constraints are optimal, then add the ad hoc request to the current release or Sprint.

#### **1.7.17.5 Communicate Changes to Release and Sprint Backlogs**

Communicate changes to release and Sprint plans to key stakeholders, requesting teams, and current Scrum delivery teams to level set expectations and needed actions on requests.

#### **1.7.17.6 Ensure Team is Aware of High Priority Needs**

Communicate priority of ad hoc requests to Scrum team developers so they may take just-in-time actions in order to analyze, design, solution, and deliver the needed ad hoc requests.

#### **1.7.17.7 Track and Report Arrival and Departure of Requests**

Ensure that all incoming ad hoc requests are visible, tracked, and known to both the requester as well as the Scrum delivery teams for proper consideration and effort as needed.

### 1.7.18 Participate in Ad Hoc Feature and Story Mapping

#### **1.7.18.1 Participate in Ad Hoc Meetings**

Participate in ad hoc meetings and technical exchanges to listen to emerging needs for new features from key stakeholders (along with the entire Scrum team and set of developers for proper input).

#### **1.7.18.2 Draft and Prepare Statement of Need**

Identify a clear statement of need, bottom line up front, assumptions and constraints, needed functional requirements, performance characteristics, and approximate timeframes or need dates.

#### **1.7.18.3 Elicit Verbal Stakeholder Need**

Have the requesting stakeholder clearly articulate the need in their own words to capture subtle nuances from the voice-of-the-customer (and make adjustments to draft statement of need for feature).

#### **1.7.18.4 Facilitate Clarification of Constraints**

Elicit questions and comments from the Scrum team, along with answers and further clarification of assumptions, constraints, and even possible solutions if stakeholder is highly technical.

#### **1.7.18.5 Facilitate Feature and Story Mapping**

Facilitate an informal working session to story map the solution to meet the assumptions and constraints of the stakeholder, along with size, timelines, risks, and proposed delivery dates.

#### **1.7.18.6 Distribute and Communicate Solution**

Distribute the story map, size, complexity, timelines, and delivery dates to the requesting stakeholder, so they may get approval from the product manager for proper consideration and timing.

#### **1.7.18.7 Adjust Release and Sprint Plans (if necessary)**

If the ad hoc feature is high priority and small, load and capacity are available, or product manager and other key stakeholders approve of the change, then add the feature to the current release.

## RELEASE WRAPUP AND CLOSURE

### 1.7.19 Prepare and Deliver Release Accomplishments

#### **1.7.19.1 Review and Analyze Release Plans**

Review and analyze current release plans, identify features that have been completed or will be completed soon, get up-to-date information from Scrum team, and identify completed work.

**1.7.19.2 Gathers Data from Scrum Team**

Gather up-to-date real-time information and data on any features, stories, tasks, ad hoc, out-of-band, significant, major, or high-priority value adding work that has been completed (or will be completed in release).

**1.7.19.3 Identifies Release Goals**

Review and record current release goals to understand Scrum team commitments, determine what features are in play, which have been completed, and whether release goals have been satisfied to-date.

**1.7.19.4 Draft Release Accomplishments**

Cull all information and data regarding completed release goals and features, and other significant, major, or out-of-band accomplishments, and prepare a draft of high-impact Scrum team release accomplishments.

**1.7.19.5 Identify Blockers**

Identify any high-impact impediments or blockers that are hindering the current increment or release, preventing accomplishment of increment or release goals, and recommended actions to resolve impediments.

**1.7.19.6 Identify Upcoming Release Goals**

Review increment and release plans, identify planned release goals, review current (adjusted) release goals from Sprint reviews and refinements, and identify the team goals for the next release.

**1.7.19.7 Verify, Distribute, and Deliver Report**

Verify release accomplishments, prepare and distribute final report, identify a reporter for release accomplishments, and participate in periodic recurring team-of-teams readouts.

**1.7.20 Measure Release and Sprint Performance****1.7.20.1 Record Planned vs. Actual Capacity**

Record planned and actual capacity in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.2 Record Planned vs. Actual Load**

Record planned and actual load in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.3 Record Planned vs. Actual Stories**

Record planned and actual stories in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.4 Record Planned vs. Actual Story Points**

Record planned and actual story points in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.5 Record Planned vs. Actual Features**

Record planned and actual features in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.6 Record Planned vs. Actual Churn**

Record planned and actual churn in ALM tools, planning pages, and other special performance measurement tools for record keeping, reporting, and alerting to key release stakeholders.

**1.7.20.7 Prepare, Verify, and Distribute Report**

Prepare, communicate, and deliver release and Sprint performance report, measurements, and deviations to key release stakeholders for proper management and contingency planning.

**1.7.21 Participate in Release Retrospectives****1.7.21.1 Participate Recurring Meetings**

Participate in recurring release retrospectives to report, identify, learn, and analyze the strengths, weaknesses, and improvements suggested by the greater team-of-teams and how they impact you.

**1.7.21.2 Access Visual Retrospective Tools**

Login to, access, or view visual release retrospective tools and workspaces to ensure you can record, review, and analyze high-impact strengths, weaknesses, and improvements across team of teams.

**1.7.21.3 Record Strengths, Weaknesses, and Improvements**

Record high impact strengths, weaknesses, and improvements preventing you from achieving release and Sprint goals or performance objectives for proper prioritization and resolution.

**1.7.21.4 Prioritize and Identify Improvements**

Vote on and prioritize the strengths, weaknesses, and improvements reported by the team-of-teams, so that they may be recognized and properly rewarded or resolved to improve performance.

**1.7.21.5 Participate in a Root Cause Analysis (if necessary)**

Participate in a root cause analysis of high priority weaknesses or improvement areas to identify and improve the key sources of impediments, risks, or performance issues.

#### **1.7.21.6 Identify Solutions to High Priority Improvements**

Brainstorm possible solutions to high impact weaknesses or improvement areas so that key stakeholders and resolution teams may have a set of possible or proper solutions to pursue.

#### **1.7.21.7 Participate in Resolving High Priority Improvements**

Volunteer to help resolve high impact weaknesses or improvement areas, especially if you are impacted by the impediment or have subject matter expertise (SME) in the domain.

### **1.8 OUTPUTS**

- 1.8.1 **Release and Sprint Objectives.** A list of bottoms up release goals and objectives generated by Scrum teams during Release and Sprint planning which are amalgamated into higher level goals and objectives.
- 1.8.2 **Release and Sprint Dependencies.** A list of key dependencies upon other Scrum teams, teams of teams, suppliers, and other internal and external teams, individuals, and organizational functions.
- 1.8.3 **Release and Sprint Risks.** A list of critical release risks associated with release features illustrating their state as resolved, owned, accepted, or mitigated (ROAM).
- 1.8.4 **Release and Sprint Capacity and Load.** The quantity of time a Scrum team can contribute to a release and Sprint in terms of staff days, along with the amount of work to be done by Scrum teams also expressed in days.
- 1.8.5 **Release and Sprint Backlogs with Features and Stories.** A list of simple features, user stories, sizes, acceptance criteria, and short descriptions in one or more Agile Application Lifecycle Management (ALM) tools.
- 1.8.6 **Release and Sprint Accomplishments.** The release and Sprint goals and objectives which Scrum teams were able to accomplish.
- 1.8.7 **Release and Sprint Performance Measurements.** Scrum team release and Sprint performance metrics and measurements such as velocity, quality, churn, predictability, planned vs. actual estimates, and other key targets.