Agile Delivery Framework (ADF)

Overview

Agile is an iterative methodology with self-directed teams and the ability to embrace change rapidly. This document summarizes the Agile Scrum process as well as defines the roles and responsibilities and milestones in a typical project.

Roles and Responsibilities

Stakeholder	Provides input into the product business and customer objectives.
Executive Sponsor	Participates in demos. Collaborates on unblocking team and continuous improvements to overall processes. Provides air-cover for team. Provides final direction on topics that are unresolved.
Product Manager	Owns the product and the product roadmap. Defines the business need and strategy. Ensures the Delivery Team fully understands the 'what' and the 'why'.
Product Owner	A role not a position. On smaller projects Product Owner and Product Manager can be the same person. Defines the problem from the customer and user point of view. Owns/Facilitates the creation and prioritization of Product and Release Backlog. Ensures the team delivers the solution completely. Connects team with customers for feedback. Can be a "one to many" scrum team relationship.
Discovery Team	Responsible for continuous innovation during a release. Consists of Product Owner, UX Developer, and Architect. Works ahead of the Delivery Team sprints.
UX Developer	Collaborates to define UX high level designs for Delivery Team. Member of the Discovery Team working one or two sprints ahead of the Delivery Team.
Architect	Insures the solution architecture is sound, sustainable, and well communicated. Also a member of the Discovery Team.
Delivery Team	Owns the business and customer problem solution. Consists of developers, testers, technical writers, etc.
Scrum Master	Helps Product Owner and Delivery Team work together efficiently and effectively. Owns/Facilitates Sprint Backlog. Unblocks Team issues. Hold Daily Standups. Facilitates Team optimization of their SDLC for the release. Coordinates the demos. Is only a "one to one" scrum team relationship. Holds sprint reflections.
Development Mgr.	Manages team staffing, skill sets, resources (hardware, software, tools, facilities), and team member career development. Works to resolve blocking issues escalated by scrum master.
Program Manager	Ensures team readiness to execute agile. Ties all the business processes together to get to market. Ensures all business processes and systems support engaged with all releases. Works with Development Manager to identify and mitigate dependencies, risks, and ensure interlocks are well understood as well as process optimization.

Scrum Process

Product Planning

Input	Actions	Who Invol'd	Output	Dur.
Product Road Map	Develop Release Themes and Epic User Stories; identify risks and develop risk mitigation stories. **Facilitator**: Product Manager	Whole Team	Prioritized Product Backlog; MVP User Stories; Decision Filters; Wireframes	1.5 d.

Release Planning

Prioritized Backlog; MVP User Stories	Complete Release Backlog User Stories; prioritize user stories based on BV (high, medium, low). **Facilitator**: Product Owner	Whole Team	Prioritized Release Backlog	0.5 d.
Release Backlog	Breakdown user stories if needed; estimate user stories using story points. **Facilitator**: Scrum Master	Delivery Team	Estimated Release Backlog	0.5 d.

Sprint Planning

Prioritized Release Backlog	Commit to what can be completed in the sprint; define 'done, done, done'; identify architectural spike need. **Facilitator**: Scrum Master	Delivery. Team	Sprint Backlog; architectural sprint backlog (optional); Definition of DONE3; Information Radiator (IR)	2-4 hrs.

Sprints

Sprint Backlog; Def. of DONE3; IR	Write test and then code; hold daily stand ups; update information radiator; continuously integrate code, refactor as required; remove tech debt.	Delivery Team	User stories that meet DONE3 criteria and are ready to be deployed	2 wks.

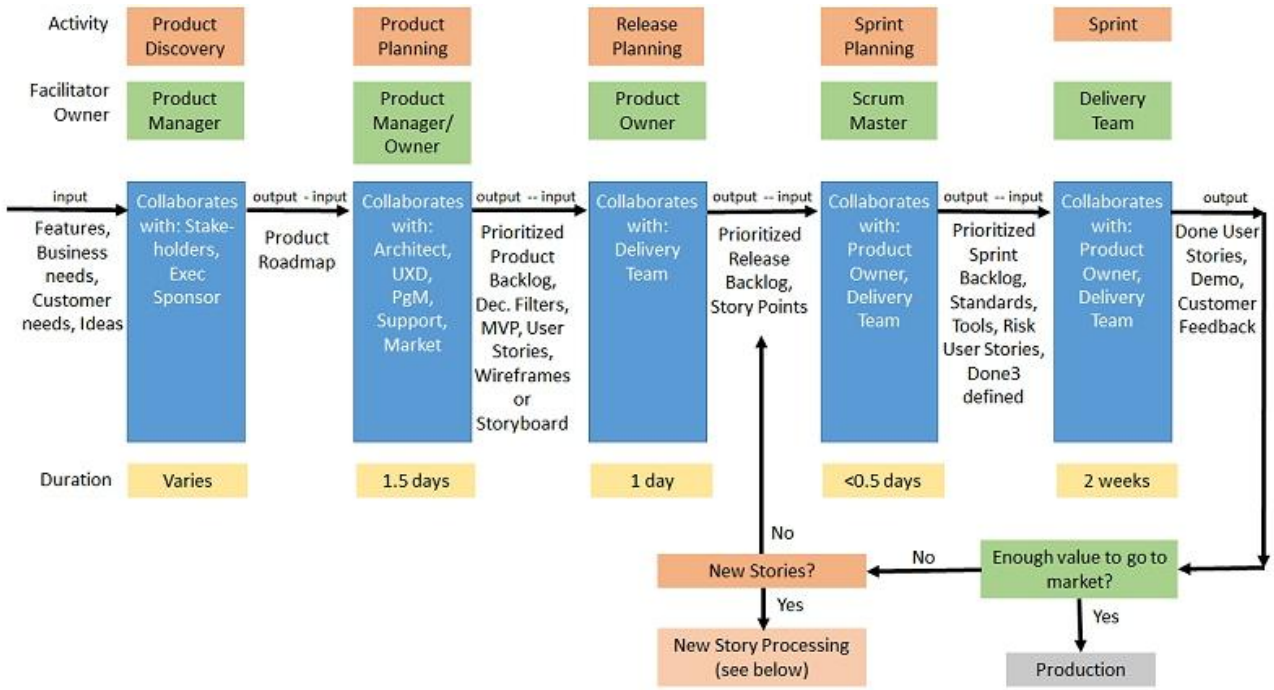
Demonstration

DONE3 User Stories	Demonstrate DONE3 user stories; get feedback from stakeholders **Facilitator**: Scrum Master	Whole Team; Customers; Exec Sponsor	Working software; not DONE3 user stories added to Release Backlog	2 hrs.

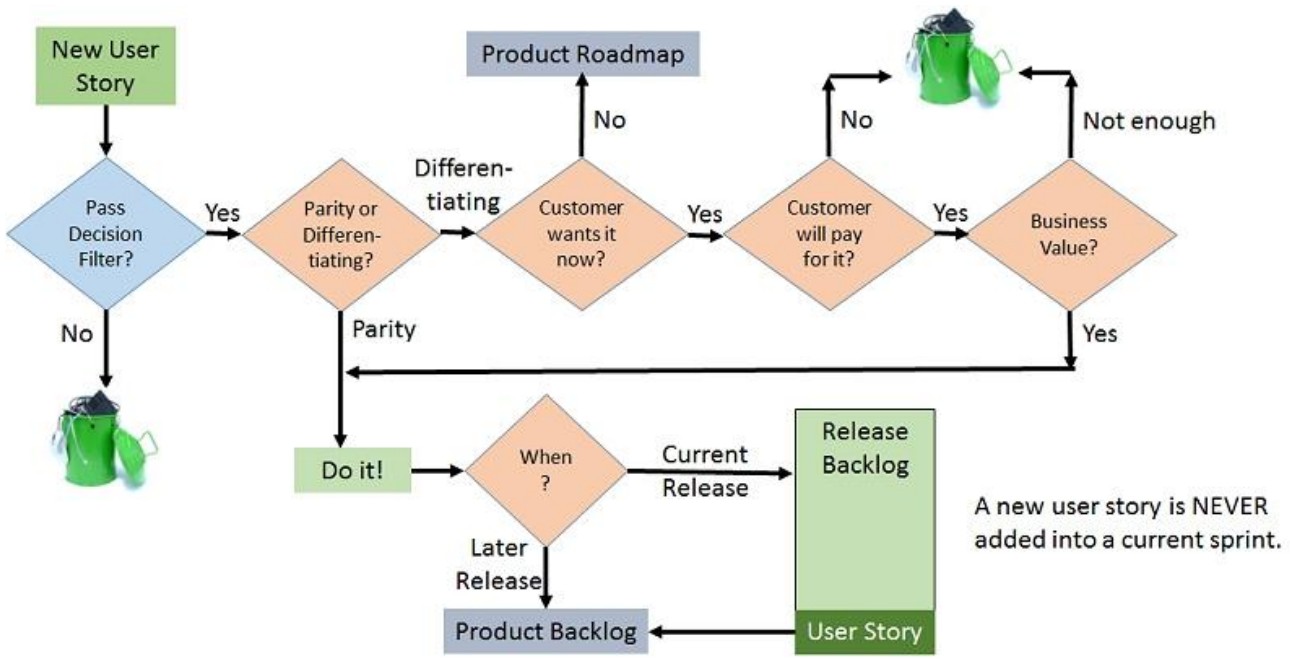
Retrospective

	Discuss process changes for next sprint. **Facilitator**: Scrum Master	Delivery Team	Changes team makes to next sprint process	2 hrs.

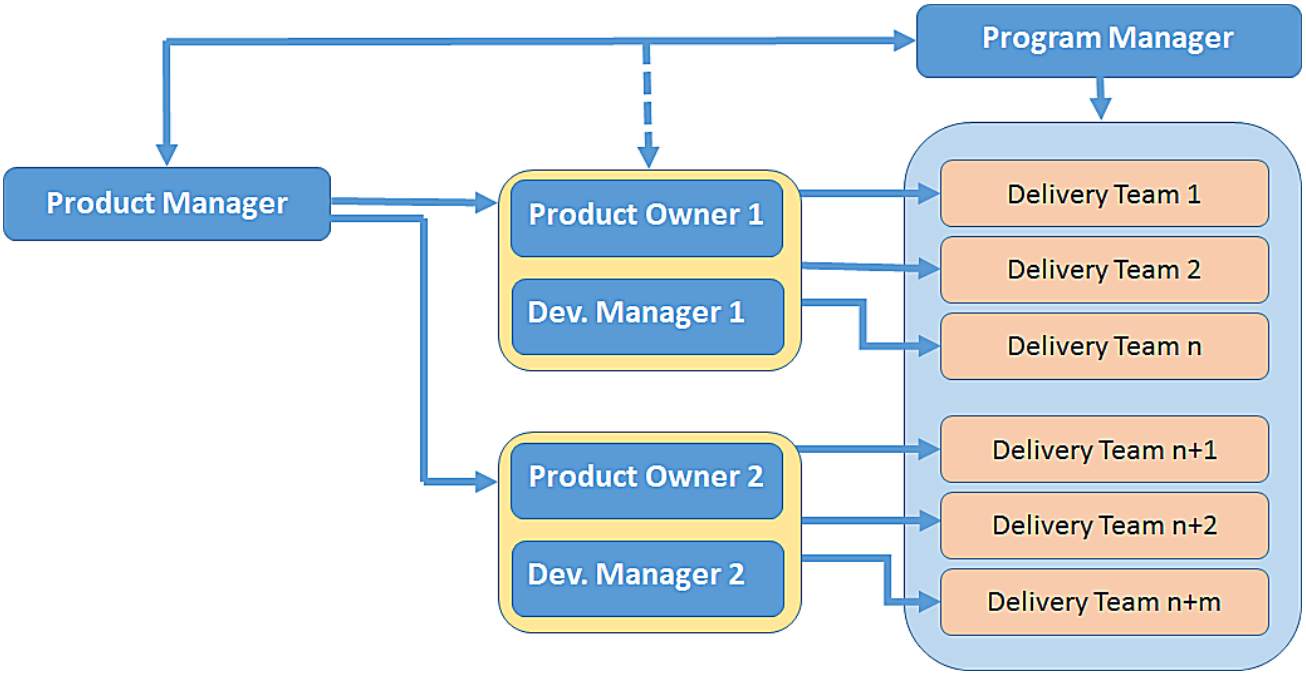
Agile Delivery Flow



Adding a New User Story to Product Backlog



Agile At Scale: Whole Delivery Team



Agile Project Readiness Assessment

In assessing a team's capability to deliver an agile project there are several areas to look at, covering at least:

- Agile Understanding
- Agile Leadership
- Planning
- Tools and technology
- Agile development process
- Skills/experience.

Note that this is an initial list. Teams should consider all appropriate risks. This is a good starting point but does not pretend to be complete. The full list of risks must be owned by the team.

The key is to ask and to keep asking the project leaders how they know that they have it everything they need to deliver. Look for concrete responses.

Step one is to use this check list to access Agile Readiness. This will indicate your gaps that need to be address before Sprint1 can begin.

Agile Understanding

- Are all team members and leaders trained? Have they attended one of the following classes?
  - Agile
  - Lean
  - Inception planning
  - Collaborative Leadership
- Do they understand the Agile & Lean philosophies and know how to act on them?
- Do they understand the ADF and know what it means to them?
- Is coaching in place and active?

Agile Leadership

- Are all members of the leadership team identified (PM, PgM, DM and supporting leadership)
- Project Owner identified and accountable

- Have they defined a lightweight governance process that will still enable team ownership?
- Are there any issues regarding the broader support of Agile by the existing management system and infrastructure

Planning

- Release planning and scope defined (via inception planning and release planning)
- What is the customer involvement in the planning cycle?
- Do they have a flexible planning process? Can they handle continuous reprioritization? (This also obviously has to apply to the business folks and processes)

Tools

- Are tools in place for key agile working in a distributed team?
- Distributed planning and estimating?
- Distributed information radiator?
- Instant communications between buddies?
- Anything else the team feels it needs
- Is an effective development environment in place including Unit test automation?
- Do they have continuous integration capability? How quickly/often they can do a full operational build?
- Do they have static code analysis tools for all of the code?
- Is there a pre-check-in test capability for the development teams?
- Do they have a fully automated test capability for all aspects of delivery? Can they / will they run a full set of tests each build? And how will they know the coverage is good?
    - Does the automation cover Unit, Functional, Systems and Non-Functional testing?
    - Do they have all the resources in place for continuous testing?

Development Process

- How long are the iterations?
- Are they doing all the things they should:
    - Involving all parties in Release Planning
    - Using Epics and User Stories for requirements
    - Using Story points for planning and tracking
    - Letting the team chose iteration content
    - Daily scrums
    - Demos each iteration
    - Information radiators for all fully testing each iteration.
- Quality Attitude
    - Are they actively managing to achieve low technical debt? How do they know they are/ are not achieving it?
    - What are their early defect removal activities? How do they know they are effective? e.g.
        - Pair Programming
        - Static code analysis
        - TDD
        - Inspections
        - Do they have tools to validate programming standards?
- By when will they have established a reliable velocity?

Skills and resources

- Team sizes and scrum team defined and staffed, including PO's and scrum masters identified?
- Are the teams fully skilled across all disciplines & domains needed?
    - Are test resources fully available for the early iterations?
    - What proportion of the team is experienced in this domain?
    - What proportion of the team is new?
    - Are the individuals committed until final delivery?
    - Are the individuals on the teams fully experienced (not just trained) in Agile?

- Are the teams entirely co-located?
    - If not, how are they going to mitigate the communications issues and hand-offs?
- Does the team have continuous access to (if not daily involvement with) the "customer"?