

Production Reference Card

Our Framework

Our Production Framework is designed to provide for incremental product development using one or more cross-functional and self-organizing teams. It provides a structure of roles, meetings, rules, and artifacts. The Framework uses fixed-length iterations, called Sprints, which are typically two – four weeks in duration. The aim of the whole team is to build a potentially shippable (properly tested) product increment each and every iteration.



The greatest potential benefit of this Framework is for complex work involving knowledge creation and collaboration, such as new product development (*i.e. Metro Firefox Browser*).

Roles & Responsibilities

Product Owner

- Responsible for product vision
- Responsible for developing the Feature Stories/Product Backlog Items
- Constantly re-prioritizes the Product Backlog
- Final arbiter of requirements questions
- Accepts or rejects each product increment
- Decides whether to ship
- Considers user and stakeholder interests
- May contribute as a team member
- Has a leadership role

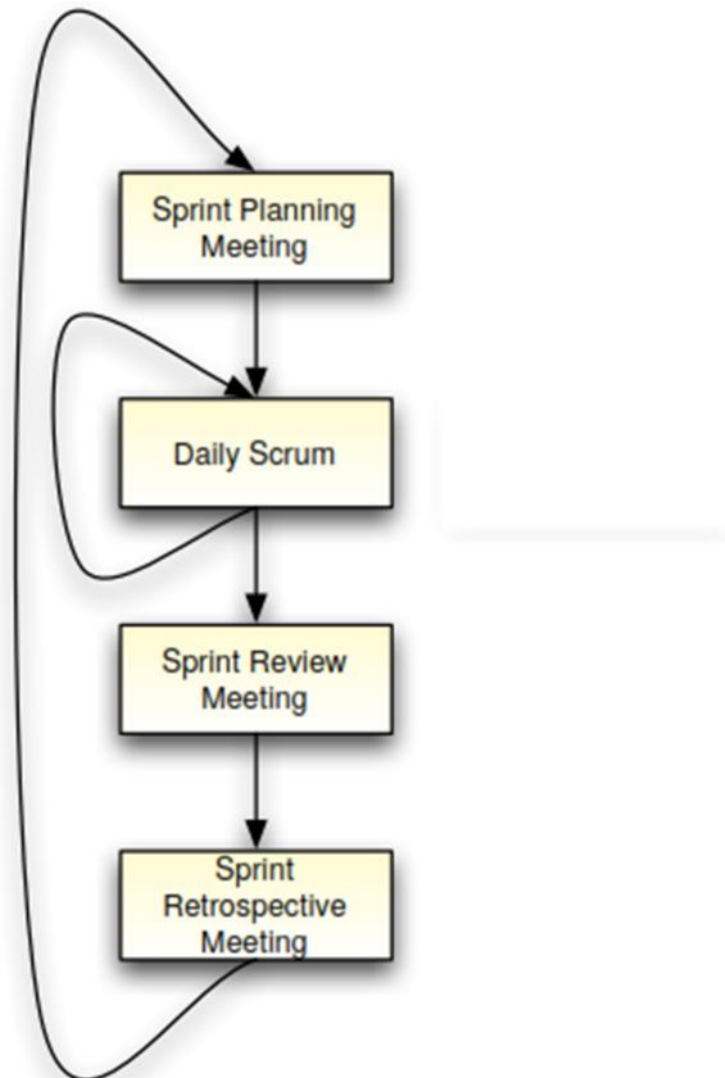
Development Team

- Cross-functional (e.g., includes members with coding, design, testing skills, and often others not traditionally part of development teams: business analysts, domain experts, etc.)
- Self-organizing / self-managing, without externally assigned roles
- Responsible for evaluating Feature Stories/Product Backlog Items for completeness and effort
- Negotiates commitments with the Product Owner, one Sprint at a time
- Has autonomy regarding how to reach commitments
- Intensely collaborative
- Most successful with long-term, full-time membership.
- Has a leadership role

Project Manager

- Facilitates the production process
- Helps resolve team and organizational impediments
- Creates an environment conducive to team self-organization
- Captures empirical data to adjust forecasts, provides updates to stakeholders and provides recommendations to achieve production goals in release plan
- Shields the team from external interference and distractions to keep production flowing
- Enforces Framework processes
- Maintains the visibility of, and updates for, all project artifacts
- Promotes improved engineering practices
- Has a leadership role

Production Pipeline



Sprint Planning Meeting

At the beginning of each Sprint, the Product Owner and team hold a Sprint Planning Meeting to negotiate which Feature Stories/Product Backlog Items they will attempt to convert to working product during the Sprint.

The Product Owner is responsible for declaring which items are the most important to the product/business and prioritizing them into the Product Backlog. The development team is responsible for estimating the work effort and selecting the amount of Feature Stories/Product Backlog Items they feel can be implement during the Sprint without accruing technical debt. The team “pulls” work from the Product Backlog to the Sprint Backlog. It is generally recommended that until a team has learned how to complete a potentially-shippable product increment each Sprint, it should reduce the amount of functionality it commits to.

Large Feature Stories/Product Backlog Items are identified as Epics and are split into smaller ones to facilitate estimating. Sometimes a subset of the team, in conjunction with the Product Owner and other stakeholders, will compose and split Product Backlog Items before involving the entire team in estimation.

Toward the end of the Sprint Planning Meeting, the team breaks the selected items into an initial list of Sprint Tasks, and makes a final commitment to do the work.

Daily Scrum and Sprint Execution

Every day at the same time and place, the Development Team members spend a total of 15 minutes reporting to each other. Each team member summarizes what he did the previous day, what he will do today, and what impediments he faces. The aim of the Daily Scrum is intended to keep all team members informed as to what is happening on the project and to disrupt old habits of working separately.

The meeting is designed to be short to keep all attendees focussed. Topics that require additional attention may be discussed by whomever is interested after every team member has reported.

The Project Manager will update any Project Artifacts as a result of the information provided at the meetings (e.g. current Sprint Task List, a Sprint Burndown Chart, an Impediments List, etc.). During a Sprint execution it is common to discover additional tasks necessary to achieve the Sprint goals or new impediments blocking the team's progress.

Sprint Review Meeting

After Sprint execution, the team holds a Sprint Review Meeting to demonstrate a live working product increment to the Product Owner and everyone else who is interested. After the demonstration, the Product Owner reviews the commitments made at the Sprint Planning Meeting and declares which items he now considers done. For example, a software item that is merely "code complete" is considered not done, because untested software isn't shippable. Incomplete items are returned to the Product Backlog and ranked according to the Product Owner's revised priorities as candidates for future Sprints.

The Project Manager helps the Product Owner and stakeholders convert their feedback to new Product Backlog Items for prioritization by the Product Owner. Often, new scope discovery outpaces the team's rate of development. If the Product Owner feels that the newly discovered scope is more important than the original expectations, new scope displaces old scope in the Product Backlog. Furthermore, QA and development team members will perform a battery of functionality tests to identify any defects which are prioritized in the Product Backlog over any new work for the upcoming Sprint.

Sprint Retrospective Meeting

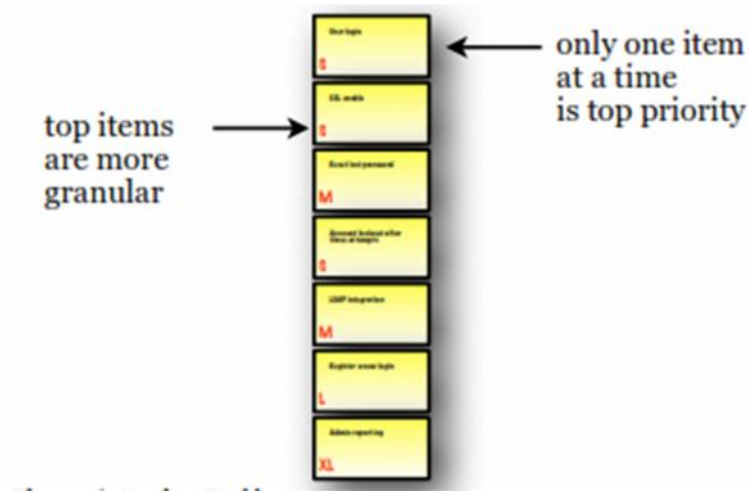
Each Sprint ends with a retrospective. At this meeting, the team reflects on its own process. They inspect their behavior and take action to adapt it for future Sprints. There are a series of steps to conducting a Sprint Retrospective Meeting: set the stage, gather data, generate insights, decide what to

do and close the retrospective. The goal are to gain a common understanding of multiple perspectives and to develop actions that will take the team to the next level.

Project Artifacts

Product Backlog

- Force-ranked list of desired functionality
- Visible to all stakeholders
- Constantly re-prioritized by the Product Owner
- Items at top are more granular than items at bottom
- Maintained throughout the life-cycle of the project
























Product Backlog Item – Feature Stories

- Specifies the what more than the how of a customer-centric feature
- Often written in User Story form
- Has a product-wide definition of done to prevent technical debt
- May have item-specific acceptance criteria
- Effort is estimated by the team, ideally in relative units (e.g., story points or ideal days)



Sprint Backlog

- Consists of committed PBIs negotiated between the team and the Product Owner during the Sprint Planning Meeting
- Scope commitment is fixed during Sprint Execution
- Initial tasks are identified by the team during Sprint Planning Meeting
- Team will discover additional tasks needed to meet the fixed scope commitment during Sprint execution
- Visible to the team
- Referenced during the Daily Scrum Meeting

Committed Backlog Items	Tasks Not Started	Tasks In Progress	Tasks Completed
	  		 
	  		
	     		
			

Sprint Task

- Specifies how to achieve the PBI's what
- Remaining effort is re-estimated daily, typically in hours
- During Sprint Execution, a point person may volunteer to be primarily responsible for a task
- Owned by the entire team; collaboration is expected



Sprint Burndown/Burnup Chart

- Indicates total remaining team task hours within one Sprint
- Re-estimated daily, thus may go up before going down
- Intended to facilitate team self-organization
- Fancy variations, such as itemizing by point person or adding trend lines, tend to reduce effectiveness at encouraging collaboration
- Seemed like a good idea in the early days of Scrum, but in practice has often been misused as a management report, inviting intervention. The ScrumMaster should discontinue use of this chart if it becomes an impediment to team self-organization.

Product Burndown/Burnup Chart

- Tracks the remaining Product Backlog effort from one Sprint to the next
- May use relative units such as Story Points for Y axis
- Depicts historical trends to adjust forecasts

Project History

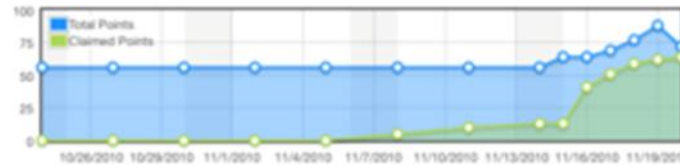


Iteration 1

Nov 13, 2010 - Nov 20, 2010

72

Points Completed

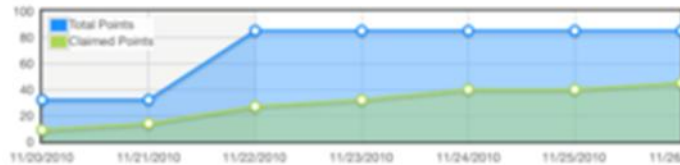


Iteration 2

Nov 20, 2010 - Nov 27, 2010

50

Points Completed

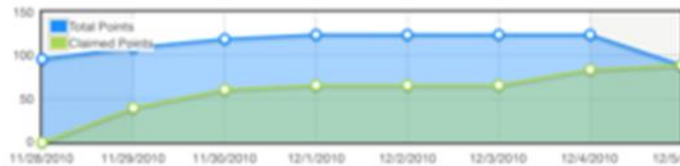


Iteration 3

Nov 27, 2010 - Dec 04, 2010

89

Points Completed

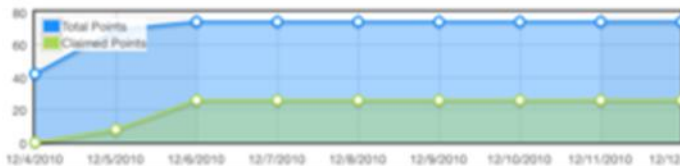


Iteration 4

Dec 04, 2010 - Dec 11, 2010

26

Points Completed



Iteration 5

Dec 12, 2010 - Jan 05, 2011



Use tags to keep track of story themes

ScrumDo

Velocity: 65
Stories: 96
Iterations Left: 17

Project Tools

- Iteration Planning
- History
- Predictions
- Planning Poker
- Export Project
- Extras
- Project Admin
- Files

Iterations

- Backlog (22)
- Iteration 1 (11)
- Iteration 2 (10)
- Iteration 3 (8)
- Iteration 4 (4)
- Iteration 5 (2)
- Iteration 6 (4)
- Iteration 7 (8)
- Iteration 8 (10)
- Iteration 9 (15)