**Guidelines for Agile Lifecycle**

Table of Contents

[1. Definitions 4](#_Toc396813045)

[2. Roles and Responsibilities 7](#_Toc396813046)

[3. Suggested Metrics For Agile Projects 8](#_Toc396813047)

[4. Best Practices 9](#_Toc396813048)

# Definitions

**Sprint**

A sprint (or iteration) is the basic unit of development in Scrum. The sprint is a "time boxed" effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, although two weeks is typical.

**User Story**

User stories are used with agile software development methodologies as the basis for defining the functions a business system must provide, and to facilitate requirements management. It captures the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail by what can be hand-written on a small paper notecard.

**Product backlog**

The product backlog is an ordered list of requirements that is maintained for a product. It consists of features, bug fixes, non-functional requirements, etc.—whatever needs to be done in order to successfully deliver a viable product. The product backlog items (PBIs) are ordered by the Product Owner based on considerations like risk, business value, dependencies, date needed, etc.

**Sprint backlog**

The sprint backlog is the list of work the Development Team must address during the next sprint. The list is derived by selecting product backlog items from the top of the product backlog until the Development Team feels it has enough work to fill the sprint.

**Release**

Releases are the result of multiple sprints of concentrated development effort, and they often mark a moment of market or business or customer impact.

**Daily Stand-ups**

Each day during the sprint, a project team communication meeting occurs. This is called a Daily Stand-up (meeting) and has specific guidelines:

All members of the team come prepared with the updates for the meeting.

The meeting starts precisely on time and would happen at the same location and same time every day.

The meeting length is set (time boxed) to 15 minutes.

During the meeting, each team member answers three questions:

What have you done since yesterday?

What are you planning to do today?

Any impediments/stumbling blocks? Any impediment/stumbling block identified in this meeting is documented by the Scrum Master and worked towards resolution outside of this meeting. No detailed discussions shall happen in this meeting.

**Sprint Review Meeting:/Sprint Demo**

Review the work that was completed and the planned work that was not completed

Present the completed work to the stakeholders (a.k.a. "the demo")

Incomplete work cannot be demonstrated

It is generally four-hour time limit

**Sprint Retrospective:**

All team members reflect on the past sprint

Make continuous process improvements

Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint?

Generally there is 3-hour time limit and it is facilitated by the Scrum Master

**Definition of Ready**

* Acceptance criteria defined in given-when-then format and well understood by teams

Story should be

* + Well Understood by teams
  + Independent
  + Acceptable
  + Estimable
  + Testable
  + Deployable in the sprint
* Assumptions, constraints and dependencies should be clearly identified/assessed (environmental, infrastructure, others)
* Basic Test data should be ready

**Definition of Done**

Every User story will have below DoD mentioned in the Description

* Developed functionality meets the acceptance criteria
* Unit Tests(Positive & Negative) are written/updated and are passed status (Percentage coverage needs to be defined at team level)
* Code is reviewed if pair programming is not followed and review comments tracked to closure
* Automated Acceptance Tests are written/updated, executed and passed (wherever applicable)
* Manual test cases (system, UI, integration, etc.) are written/updated, executed and passed for the User Stories wherever applicable
* Necessary comments are updated if any of the above criteria is not applicable
* Source code is checked in code repository with appropriate comments added.(e.g. User story no, pair involved etc.)
* User stories/tasks is updated with appropriate status for
* Defects related to the user story logged, fixed and accepted.

# Roles and Responsibilities

|  |  |
| --- | --- |
| **Product Owner** | * Responsible for priorities for sprints. * Responsible for pulling in stakeholders for Sprint reviews or planning as needed * Responsible for verification of functional details for all owned user stories while grooming. * Responsible for acceptance of owned user stories * Responsible to make sure all PSI user stories are in ready state * Responsible for back and forth communication between IT and business for open questions * Responsible to make sure that there are least number of unknowns to the team on the day of sprint planning |
| **Scrum Master** | * Responsible for gaining team’s buy in on user stories * Responsible to make sure that team is working on right priorities * Responsible to communicate back if team is over or under allocated * Helps PO in defining Acceptance Criteria * Communicate risks to the PO * Collaborate with PO on scope   **For The Team**   * Remove road blocks of the team * Communicate risks & dependencies with the team (SOS-Scrum of Scrum) * Helps the team in mitigating the risks * Provide constructive feedback and help individual to work on it * Boost up the moral of the team   **Process Implementation & Improvements**   * Drive different scrum framework ceremonies [Stand-ups, SOS, Retrospectives, Sprint Planning, et al] * Monitor & control the health of sprint using Metrics * Ensure that the action items are identified for every sprint and tracked to closure * Ensure that the teams follow "Definition of Ready" & "Definition of Done" * Escalate to the appropriate stakeholders in case there is any deviation in the process risks * Identify the risks and raise it * Document the retrospective points and update the presentation |
| **Scrum Team** | * Responsible for meeting the goal to which the team wholly commits every Sprint * Estimate ((story pointing)) user story * Design and implement the User Stories * Developers will write the code, unit tests scenarios/test cases (positives as well as negatives) * Developer collaborate with the product owner and testers to make sure the right code is being developed * Testers work in parallel with developers to write acceptance test cases * Testers will interface with the product owner and developers to confirm that the code and acceptance tests reflect the desired functionality * Testers will execute acceptance tests * Testers will write manual test cases and maintain the test cases * Conduct reviews and Log defects and track till closure * Participate in daily standup meetings, * Participate in sprint demo and retrospective * Participate in release and sprint planning meetings * Estimate (story pointing) sprint tasks * Prepare for next sprints * Perform Development / Support activities of medium complexity as per project plan & schedule. |

# Suggested Metrics For Agile Projects

* Say do Ratio: No of User Stories Accepted Vs No of User Stories Committed
* Velocity: No of User story points delivered per Sprint
* Throughput: No of Work items delivered per Sprint
* No of Defect injected per sprint
* Cycle Time: Time taken for story to move from in progress to accepted state
* Lead Time: time take for story to move from backlog to accepted state

**Refer Agile Metrics presentation for more details**

For Monitoring

Release Burn-up charts

# Best Practices

1. Team members are collocated with required roles.
2. Release Planning in Agile Projects

|  |  |  |
| --- | --- | --- |
| Product Backlog | Create Release Plan  - No of sprints  - High Level estimates (e.g. T-shirt sizing etc)  - Features to be covered  - Priorities  - Release schedule  - Cost  - Identify risks along with mitigation and contingency plans  Gate 1 approval would be requested after planning phase | Release Plan |

1. In case there are many scrum teams working on the same product teams can derive their Team Backlogs from Product Backlog and then get Sprint Backlog from team backlog
2. Defect Review Meeting

* Prioritize defects based on business priorities and requirements
* Reducing defects

1. Backlog Grooming

* Story pointing can be done when stories are being explained and groomed
* It can save efforts
* QA should be involved in all the sprint planning and involve in definition of acceptance criteria

1. QA focus could be more in automation testing
2. Soft Review

* One person in team will be dedicated reviewer who will monitor commits done by the team. He will provide suggestions and feedback for user story.

1. PSI Planning ahead of Time

|  |  |  |
| --- | --- | --- |
| **Input** | **Activity** | **Output** |
| **3 PSI Out**  Business Problem Statement  Prioritization / Backlog Grooming(6 months ahead of time) | * “Placeholder” user story is written   + Product Owner identified   + First draft of “feature problem statement”     - As a <role>, I need to <behavior> so that <business value>.   + Team(s) impacted identified   + Dependencies identified   + No acceptance criteria   + No value stream/team identified yet   + Release target identified   + High level estimation assign to the feature   + System architecture needs to be considered | Marketing one pager (optional)  Technical one pager / Technical High Level Solution Document |
| 2 PSI Out(4 months ahead of time) | * Full user story is written   + Acceptance criteria written   + Value Stream/Team identified   + Mock-ups completed   + Non-functional requirements documented   + Business owner signs off on user story * System architecture should be finalized | Use Case Documents  UI Mockups  Technical specifications documents  Impact Analysis  Define Estimation |
| 1 PSI Out(2 months ahead of time) | * If feature warrants it, feature is split out into multiple “child” user stories   + All “child” user stories given the final point estimate * Impact analysis completed * Test plan created (and put into user story) | Customer Support Plan (optional)  Go to Market Plan (optional)  System Integration & Testing  User Stories (for sprints)  Ready for PSI Planning |
| PSI Planning | * User stories selected for the current PSI based on business priorities   + Dependencies to all teams identified and sprint plans are aligned   + Capacity is considered and stories are selected * Sprint plans are created   + Tasks identified and task owners assigned | Team assignments (Tasks identified and owners assigned)  Approval of the plan by the team |
| PSI Ready | * User stories are worked on by teams   + User stories are accepted by product owner   + If something was missed in user story creation, another user story can be created for future development * Once stories are completed, tested, and accepted, changes are delivered to customer * Agile ceremonies (demo, retro, sprint plan, business UAT) | Delivery to the Customer |

**Roles involved:**

3 PSI Out:

Product Manager/ Business Lead – Marketing One Pager

Product Owner / Business Analyst – High Level Estimation

Architect / Tech Lead – Technical One Pager

2 PSI Out:

Product Owner / Business Analyst - Use case Documents, UI Mockups?

Architect / Tech Lead - Technical specification documents, Impact Analysis

1 PSI Out:

Product Manager/ Business Lead - Customer Support Plan, Go to Market Plan

Tester - System Integration & Testing

Product Owner / Business Analyst - User Stories

Scrum Master – Capacity Planning

PSI Planning:

Team – Define estimation, task defined and test owner defined

Scrum Master – Presenting final PSI Plan

Product Manager /Business Lead – Plan Approval