ePATH USA
Premier IT Services

Sprint Development Procedure

Version No: 1.0

Date: May 03, 2017

Revision History

Version	Date	Prepared by / Modified by	Change Summary	Approved By	Approved On
1.0	May 03, 2017	Shiva Meduri	First Draft Made	Anitha T G	May 18, 2017

Table of Contents

1. Purpose

The purpose of the document is to define the activities related to sprint development. Using this process, the team should be able to meet the required functionality and meet the definition of done (DoD) at the end of sprint.

2. Scope

This process is applicable for the projects that are executed in ePathUSA.

3. Entry and Inputs

3.1. Entry Criteria

This process will start as soon as the project planning is completed.

3.2. Inputs

- Project Management Plan
- Contract
- Statement of Work
- Product Backlog if exists

4. Activities

4.1. Create or Update Product Backlog

- Discuss high level requirements/scope/vision of the product
- Identify the requirements providers and update them in the project management plan under external stakeholders
- Document the high-level requirements (functional / non-functional) in the product backlog by discussing with product owner or users. The scrum master can use a separate document for Requirements Development in case the Requirements itself is being considered as a separate sprint.
- Gather the user stories through appropriate techniques such as
  - Brain storming
  - Workshops
  - Observation
  - Discussions etc.,
- Understand and clarify the user stories that are incomplete or missing clarity.

- Discuss the user stories with the team for their feedback. Based on the feedback, perform any of the tasks below
    - Add, remove, modify the stories
    - Split the user stories into smaller items
    - Merging Items into larger items
- Assess the story clarity to ensure that the stories are appropriately defined without any ambiguity and missing details.
- Use below criteria to assess the user story clarity (refer to the User Story Review checklist for the details criteria
    - Definition of user story is comprehensive
    - Defined acceptance criteria
    - Granularity of the user story
    - Any dependency, interface or non-functional requirements associated etc.,
- Assess the clarity of user stories and if required refine the user story description in discussion with product owner.
- User stories will be prioritized by the Product Owner based on the business value and his business needs.
- Identify and define the acceptance criteria or Definition of Done and update in project workbook against release and sprint.
- Team will estimate the user story size using story points
- Scrum Mater should list the dependencies and risks associated to the backlog items in the project plan or risk tracker as appropriate.

4.2. Create Sprint Backlog

- Understand and analyze the user stories from technical aspects such as impacted modules, complexity, interfacing components etc.,
- Re-evaluate the story for any inputs from the product owner before starting the sprint, communicate to the product owner and get clarity on the blockers as appropriate.
- Communicate assumptions dependencies related to stories for the current sprint and update them in project workbook.
- Update the traceability matrix with respect to the dependent requirements
- As required the team can prepare prototype / wire frames to get more clarity on the requirements and discuss with product owner to be in sync.
- Selected user stories should be agreed with product owner before executing the sprint.

4.3. Design

- Evaluate Design alternatives, analyze and select the appropriate Design methodology suitable for the project. The optimal solution should be selected based on the evaluation.
- Operating concepts and scenarios should be established for each of the alternate solutions
- While designing the project can consider the following elements
    - Modularity, reliability, performance
    - Usability, Security, Quality
- Design can be developed using
    - High level or Architectural Design
    - Detailed Design/Program specification
- Design should involve breaking of the User Stories into modules with information on Interface Requirements between the modules.
- Detail Design typically involves:
    - Data elements
    - Data structures
    - Input
    - Output
    - Processing details of each component
    - State machine
- Design should be reviewed by the identified reviewers.
- The traceability should be established from User Stories to Design modules including interfaces. Update the Requirement Traceability Matrix.
- The product owner may also be involved in the review and approval of Design document, if required.
- The reviewed and approved copy of Design document should be placed under Configuration Management.

4.4. Coding

- Identify and deploy the agile practices as applicable, some of practices include Refactoring, Pair Programming, Test Driven Development etc.,
- Code is developed based on the program logic, input, output and associated data structure.
- Consider the automatic code generation, reusability, coding patterns, coding standards etc., while developing the code.
- Coding done using Coding Standards and Naming Conventions established for the project
- Ensure that the code is simple and maintainable. The code should be compiled and made free of syntax errors

- Perform review and close all review comments.
- Source Code along with build instruction and user documentation (if any) should be placed under Configuration Management.
- Update the traceability Matrix.

4.5. Code Review

- Code should be reviewed as per the review criteria
- The scrum master will decide the scope and type of review based on the project needs in the project management plan
- Check-in the code after fixing all review comments and verified.

4.6. Unit Testing

- Identify the unit test cases including the expected results, test steps any data requirements.
- Execute Test Scripts / Test Cases
- Fix code for failed test cases and re-test
- Errors will be logged into the defect report.
- Modified code will be released for next level of testing as per the test strategy

4.7. Build and Integration

- Scrum Master with the team determines the Integration strategy and accordingly the user stories get prioritized during the sprint planning meeting.
- Team identifies the components to be integrated along with the integration test cases
- Team should review and manage interface compatibility components, interface descriptions and other data to ensure coverage of all interfaces in the design document.
- Team verifies integration environment, and build is in place and ready to use.
- Recurring build is created and appropriately sanity checks are done to ensure the integrity
- Fix all build failures, re-test, re-build
- Deliver the code for system and integration testing

4.8. Testing

- Set up the necessary infrastructure
- Evaluate the test results against the acceptance criteria.
- Measure the code coverage if available through tools and take appropriate action.
- Log defects reported
- Retest the bug fixes and track the status to closure
- Revisit the test Suite during execution and modify as appropriate

4.9. Customer Review / Demo

- Demonstrate the features developed during the sprint
- Product owner should review and provide acceptance based on the definition of done
- In case if the feedback is related to a new user story include the same in product backlog
- If the feedback is related to the story developed, then documented and track to closure
- Customer can test the features and provide the feedback to the development team.

4.10. User Acceptance Testing

- Scrum Master should plan for UAT as per the release plan
- Before the release the team should provide the internal Tested Build to customer
- Customer will provide any issues observed during the UAT
- The team will analyze the issues, fix the issue, and create new build for re-testing as required.
- The team will provide the complete support during the UAT
- Document the customer reported issues in the test report under defect log

4.11. Production Release

- Update the user manuals or any other user documentation as expected by the product owner.
- Team should develop deployment and installation documents.
- Create the release deployment package and tag it with a name
- Prepare the release note and communicate it with the required stakeholders
- Get the sign off from the product owner
- Known bugs, if any should be documented in the Release Notes

5. Exit Criteria and Outputs

5.1. Exit Criteria

An exit criterion for this process is when the sprint / release is delivered to the customer.

5.2. Outputs

- Product Backlog
- Sprint Backlog
- Design
- Code
- Test Cases
- Test Plan

- Test Results
- Review Records
- Defect logs
- Release Note

6. References

- **Templates**
  - Project Management Plan
  - Review and Test Log
  - Project Workbook (Traceability Matrix)
- **Procedures**
  - Spring Management Procedure
  - ePathUSA_Policy
- **Guidelines**
  - Agile Methodology Guidelines