Our Software Development Process

On this page

Overview

Currently the IT Application Development team uses a modified Agile Scrum methodology for software development. Modifications are generally around being able to satisfy the business need for coming up with ballpark figures before fully defining the requirements. As a result our road maps are based on those estimates rather on estimates on actual stories. We've adopted our own version of Agile which should be viewed as a changing/evolving process that we scale based on the size, length and complexity of a project. Our process consists of design and story writing sessions with business owners/stakeholders, product management and if necessary technical lead. This is followed by sprint review with QA and technical lead, followed by a tech sprint review by developers. Once everyone is clear on what needs to be done we go into a sprint planning session where developers estimate stories and we decide the full length of the sprint. In other words based on the time we have available we decide which stories we keep in the sprint in question and which stories we push to the next sprint. If appropriate we may decide to shorten or lengthen the sprint in question.

Collecting Requirements (Story writing)

To be clear the requirement gathering falls outside of the Application Development teams area of responsibility. This section is really about refining those requirements so that the developers can translate them into code.

Standard practice for writing the stories is well documented from various online sources. The basic concept is fairly simple and the idea is to answer three questions within a user story, who, what, why. Once the basic story has been written it should be placed in JIRA in the project's backlog. It should then be refined further by meeting with team leads (Technical (Dev's), Quality (QA's) and Business Analysis (PO's)) and author, for initial screening. If it passes the refinement session it should be brought in front of the developers for technical review for further refinement. Feedback collected from these meetings should be brought again in front of the business owners of the requirement in question. For full description of the meetings see the "Suggested Meetings" section below.

✅ **User stories** are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template:

As a < type of user >, I want < some goal > so that < some reason >.

Sourced from Mountain Goat Software site.

Meetings, Roles and Responsibilities

There are four recommended ongoing meetings for any given software project.

1. **Requirements Gathering Meeting**
    a. **Attendees:** Subject matter experts (SME), product owner/manager (PO)
    b. **Goals**: In these meetings, the story should be written and the PO should be able to identify if input from UX/UI designer is needed going forward. If this input is needed the designer should be invited to the refinement meeting or met with separately before the refinement meeting. Alternatively, PO can produce wire-frames if changes are minor, adding a new field, rather than major, creating the new form(s) or adding multiple fields.
    c. **Roles and Responsibilities:** BO and PO are required for this meeting and if either can't participate the requirement in question should not proceed further.
        i. Business owner/business process expert should come to these meetings prepared to talk about the business process or function for which the requirement refers to. It's recommended that a business process diagram is made before or during this meeting to document the current process around the requirement, if applicable. If this is something completely new then a diagram of the desired process should be prepared. If the BO/BPE is not able to produce these beforehand it is highly recommended that this is captured during this meeting. It should be noted that not following these recommended guidelines will result in a delay in implementing the requirement in question. This will occur due to a higher number of iterations needed in order to both refine the requirement as well as implemented it if the refinement recommendation guidelines were not followed. Example: no process diagram for the current process, no diagram gets created in refinement, developers have to implement in best judgment which may result in a need to iterate an increased number of times in order to actually capture the desired functionality.
    d. **Output:** First draft of user stories with the basic Who? What? Why? answered as described above with subsequent iterations of this meeting resulting in wireframes (if applicable) and business process diagrams (if applicable).

2. **Requirements Refinement Meeting**
    a. **Attendees:** Developer lead, QA lead, PO's (and all BA's that have a requirement that is up for discussion), PO, UI/UX designer (if applicable), Development Manager (optional), QA Manager (optional).
    b. **Goals:** Go over the stories in question and fill out any details that the leads may identify. In general, these might include missed impact on other aspects of the application or some missed detail in the process or UI. The end result might be that the story needs to go back to the SME in order to answer questions raised by the leads and depending on the timing the story may end up getting moved to a later sprint than originally intended.
    c. **Roles and Responsibilities:** PO leads this meeting but the meeting should be organized by the

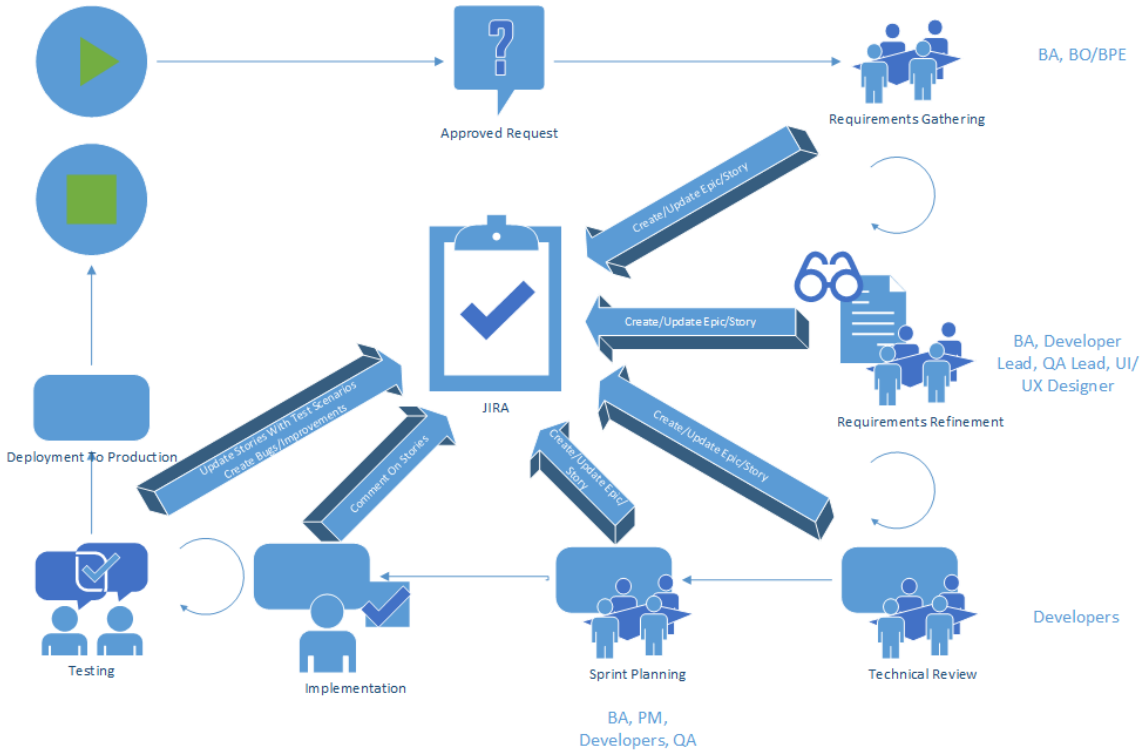
Scrum Master (SM). PO would walk the rest of the group through the business requirements. Developer lead should validate any assumptions made and focus on understanding the functional requirements in order to be able to translate them to technical specifications. QA lead should validate test-ability of the requirements. UI/UX designer would identify what needs to be designed in terms of the UI/UX and provide feedback on any immediate issues.

d. **Output:** Refined requirements potentially further broken down into further user stories. Questions documented for follow up with business. Requirments may need to go back to the business and go through another round of the requirement gathering meeting.

3. **Technical Review**
    a. **Attendees:** Developer lead, developers on the project, infrastructure dev ops lead (if applicable)
    b. **Goals:** Identify items that need architectural design and assign a task to draft those. Identify new items that conflict with what is already implemented and may require further functional considerations. Draw up questions on stories and send them back for further refinement by BA's, BO's and UI/UX designers.
    c. **Roles and Responsibilities:** Developer lead would organize this meeting and if possible identify if infrastructure, data architect or any other technical resource needs to be invited to the meeting. The group would identify items that need to have further considerations or need to be architected. As an example, if there is a requirement that impacts the data model that impacts the ability to connect to other systems or do cross-system reporting, data architect would be needed to think through the solution and document the design. If it's a process change within the system, it may need a new sequence diagram or an update to existing one. If it involves scale expansion, infrastructure lead needs to come up with new design and plan for connecting to other resources on the network, potentially prepare for load testing.
    d. **Output:** Documentation and design tasks identified and assigned. A decision on whether the story can be implemented or needs further refinement.

4. **Sprint Planning**
    a. **Attendees**: PO, SM, QA, developers, others can be added optionally
    b. **Goals**: Set sprint duration, from what date to what date. Set sprint capacity, how many developer hours and QA hours do we have for the sprint. Set sprint goals, what do we want to accomplish with the sprint and will we go to release it. Developers will estimate the effort in half-hour increments for each story or task. Ideally, you want to put estimates on all stories that are ready to be estimated, not just the ones you intend to include in the next sprint. This way you have stories in the backlog that you can pull into the sprint if developers overestimated, this will also make next sprint planning quicker as you are getting a better-developed backlog allowing PO's to plan more accurately.
    c. **Roles and Responsibilities:** SM would organize and lead this meeting. Developers are responsible for estimating the stories and tasks and there should be an agreement between them as to what the estimate is.
    d. **Output**: Sprint defined with clear start and end dates as well as scope.

Meeting Focused Process Diagram



Responsibilities by Role

Responsibility	SME	PO	Scrum Master	Dev	QA	Infra	UI/UX
Provide requirements	X						
Collect requirements as user stories		X					
▪ Design UI wire-frames and UX behavior							X
▪ Write acceptance tests					X		
Prioritize user stories	X						

Organize the backlog		X					
Schedule and facilitate Scrum meetings			X				
Administer the Scrum tool			X				
Coach team and ensure process compliance			X				
Technical reviews				X			
Document technical artifacts				X		X	
Preform tests					X		
Demo the sprint				X			
Refine requirements		X					

Suggested Role Mapping to Common IRC Job Titles

IRC Job	Scrum Role
Project Manager (PM)	Scrum Master (SM)
Business Analyst (BA)	Product Owner (PO)
Product Manager	Product Owner (PO)
Business Owner (BO)	Subject Matter Expert (SME)
Business Process Expert (BPE)	Subject Matter Expert (SME)
Any other business position that may use the product being developed	Subject Matter Expert (SME)

Scrum Sample Project Calendar

Related Articles and Links

- **Atlassian Scrum Guide - more in-depth overview of Scrum**
- ***How to come up with a road map within an agile process*.**
- ***Story writing (sourced from Mountain Goat Software site)***