**Development Working Process**

**Nov 2019**

# STAKEHOLDERS

Development team at SIM-VN Office & SIM-MED office, POs, BAs, Scrum masters, Developers, Testers, etc.

# OVERVIEW

This document provides a general guideline for development team members to understand and follow a standard development process, to ensure development activities running smoothly; and by that, we hope we can achieve better quality & productivity of work and less pressure for life.

# PROCESS

## Methodology

We’re using Agile/Scrum and we will keep it as-is.

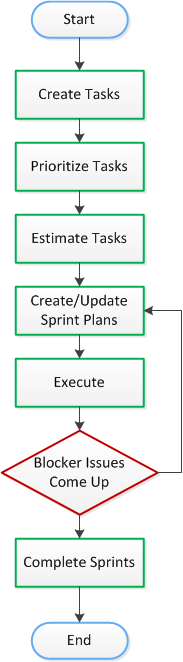
## General things to keep

* Daily scrum (15-min standup meetings)
* Daily sprint review
* Retrospective and planning for new sprint

## 

## Tasks creation and planning

### Sprint activities process



* Follow priorities and ranking: Blocker > Critical > Major > Minor
* New Blockers will automatically be added to the current sprint, meaning that it will be processed **immediately** unless we have other blockers with higher ranking.
* Completed issues of a sprint (including Blocker ones) will be deployed to production by the end of sprint, after got 'Ready Production' status from PO.
* In case there are blockers that cannot be completed in a sprint (even after sprint extension if any) and have to be delayed to next sprints, those Blockers will automatically be understood as Hot fix requests (see 2. Hot fixes process), and in that case, technical team will confirm and get approval for those hot fixes deployment schedule from PO.

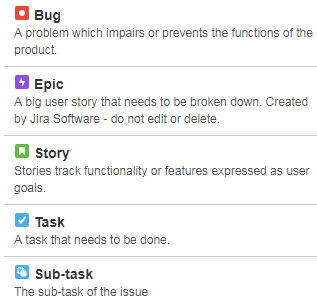
2. Hot fixes process:

* If a blocker is requested to be hot fixed, meaning that it will be deployed to production as soon as possible when it is completed (status 'Ready Production' from PO), and do not need to wait until the end of sprint.
* Of course we will need some testing effort for every hot fix, although it should be much smaller than a full regression test.

## Task Types

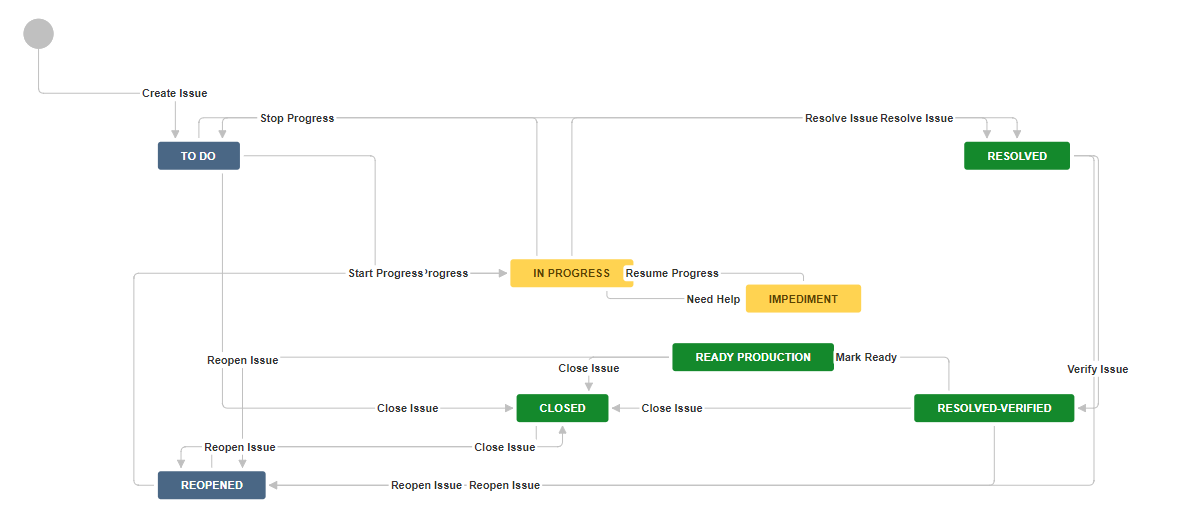
Epic > Story > Task (Bug) > Sub-Task

*Notes: I usually use just Epic, Task and sometimes Sub-Tasks*

**

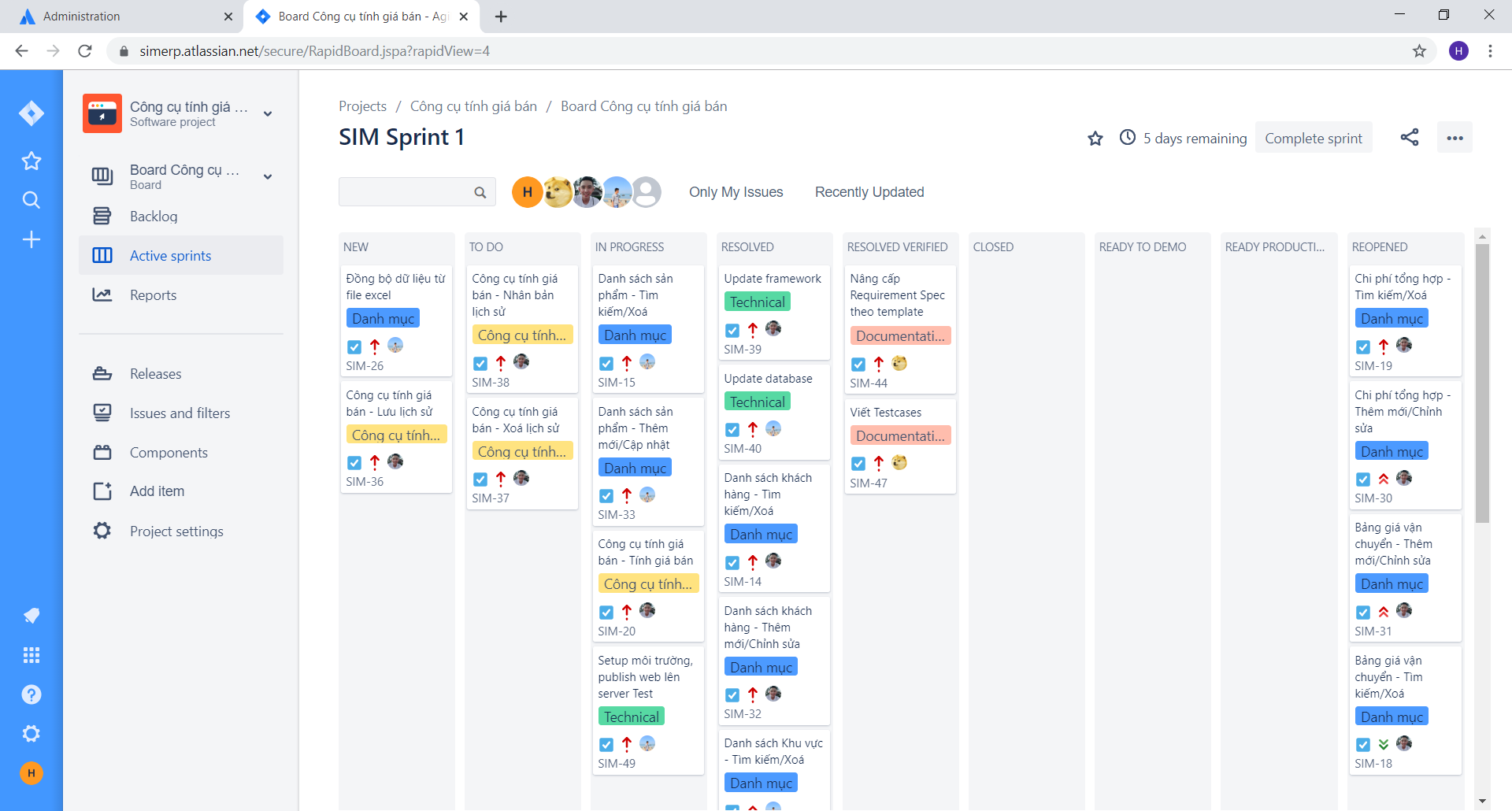
## Tasks processing

* Issue workflow



* Statuses:

|  |  |
| --- | --- |
| **Status** | **Description** |
| TO DO | Progress on issue has not been started. |
| REOPENED | Issue marked as resolved intensively but actually not, reviewers reopen the issue so that work on the issue will be continued. |
| IN PROGRESS | Someone working on the issue. |
| IMPEDIMENT | Assignee needs help from others. |
| RESOLVED | Assignee completed the task, but it is not officially passed expected testing/reviewing round. |
| RESOLVED-VERIFIED | Issue completed and verified e.g. A feature has been implemented and tested-passed on DemoCenter, ready for UAT testing |
| READY PRODUCTION | Issue has been passed UAT testing and qualified for production deployment.   1. If it is currently assigned to PO team, it means that it is pending production deployment. 2. If it is currently assigned to Development team, it means that PO team requests to deploy this feature to production environment. |
| CLOSED | Task has been completed successfully or reporters decide not to do it anymore. |



* Rules (mostly for technical team)
  + Issues must be logged work daily for tracking and status updating purpose.
  + As soon as the assignee realize that he needs more time for the task or less time for the task (about ± 30%), he needs to notify the Scrum master (team lead), do not wait until the last minutes.
  + If it’s a technical item which requires new technical solution and/or a technical decision needs to be made, please make sure you write down in the ticket about technical details of the solution/decision so that others can review and provide input when applicable.
  + When you commit source code or document for a ticket to SVN repository, make sure to mention the ticket ID in your commit message e.g. “Abc xyz **1-118** ...”, by this, your committed revision info will appear within the ticket on Jira.
  + When a developer or a QC finish his job, he needs to capture screenshots/videos as evidence for the job, this is just to make sure we don’t miss the testing/verifying steps.
  + Especially for New UI version development, TCs list for a feature should be available before the implementation job for that feature is done. Developer of that feature should use the TCs list to self-check his feature and capture screenshots/videos accordingly. Then the QC should also use that TCs list to report test results.

## Priorities

|  |  |  |
| --- | --- | --- |
|  | Hot Fix | To production immediately |
|  | Blocker | This problem will block progress. |
|  | Critical | Critical Serious problem that could block progress. |
|  | Major | Has the potential to affect progress |
|  | Minor | Minor problem or easily worked around |

## 

## Tools

Suggested tools:

* Project Management: Jira – Cloud based
* Documentation: Google apps
* Storage and revision control (source code, documents): SVN – self hosting

# TIPS

## How should we start a day?

1. View the backlog for your assigned tasks
2. View list of Impediment tasks that might need our support
3. Check Jira email especially ones which have mentioned us.

## Useful quick filters

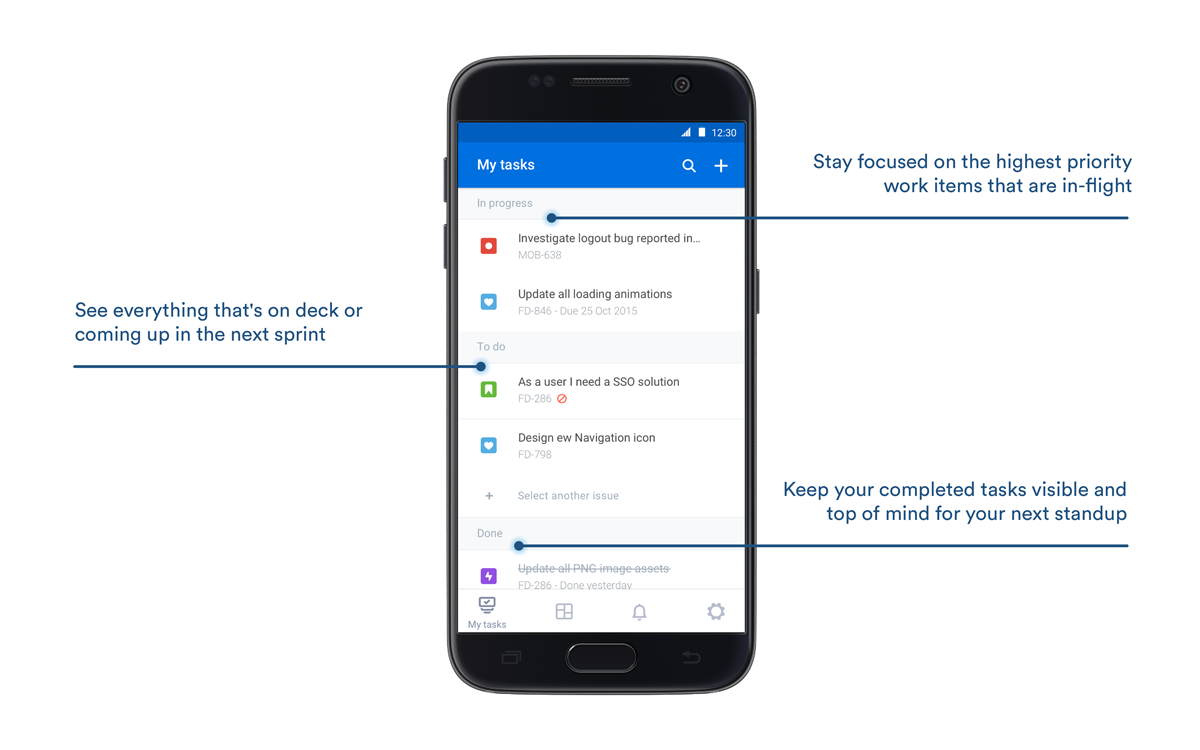
These are available on the Jira Board

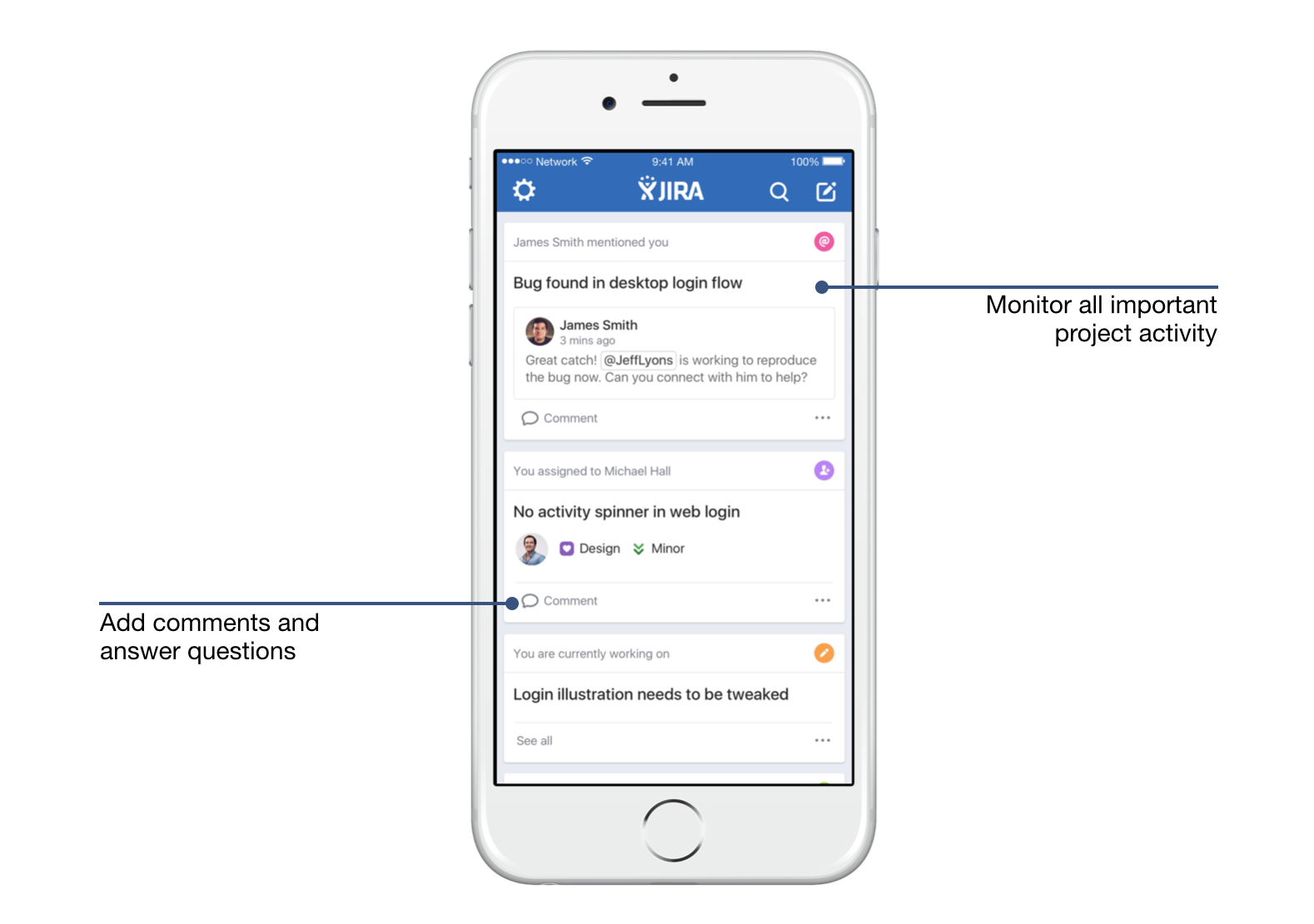
* Impediment Issues: where we can see who needs help from whom.
* Only My Issues: view our own assigned issues
* Recently Updated
* Individual filters

## 

## Jira on Mobile

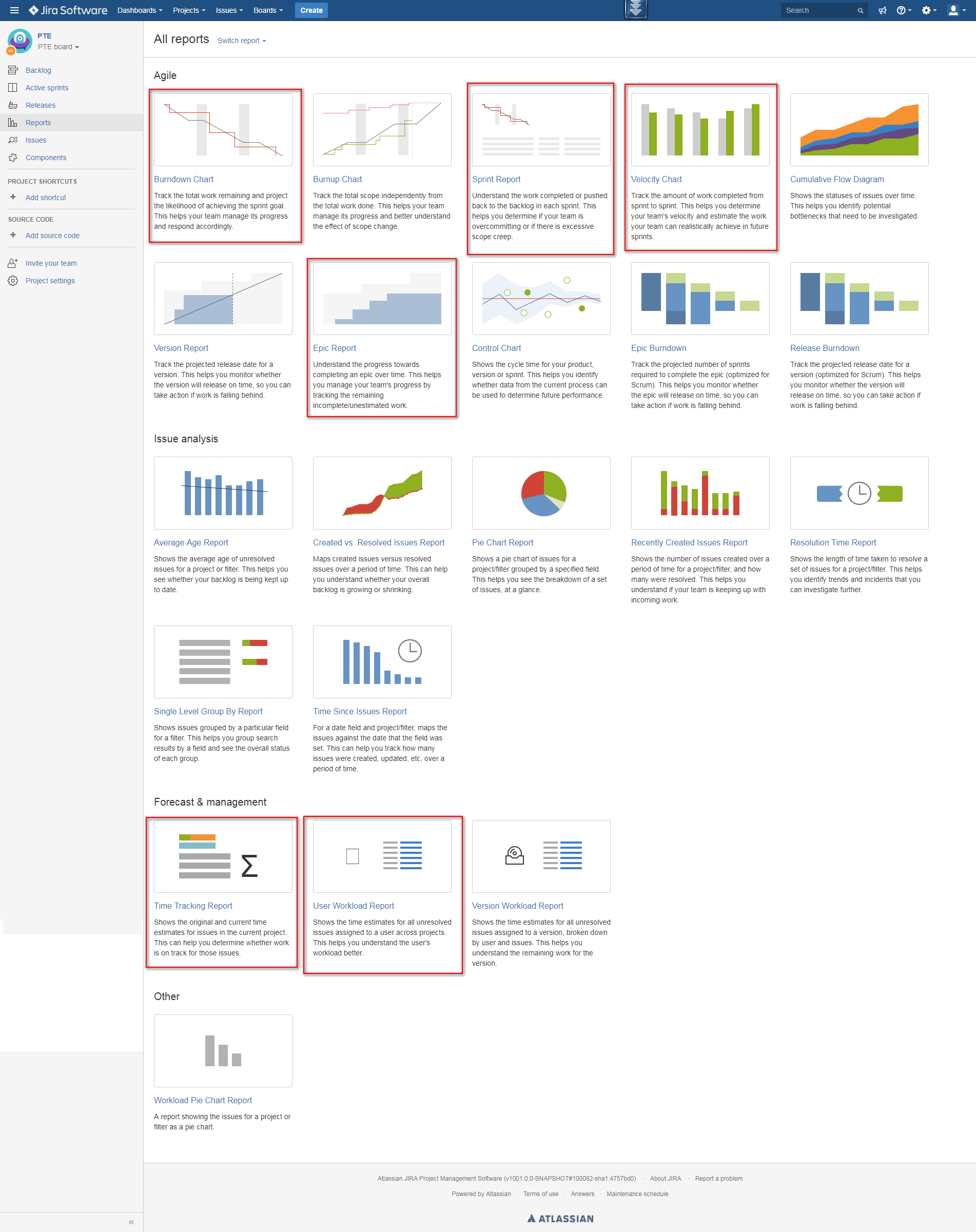
In addition to working with Jira desktop version, we should also use Jira on Mobile as it would be convenient and save time in some cases.





# Common Views and Reports

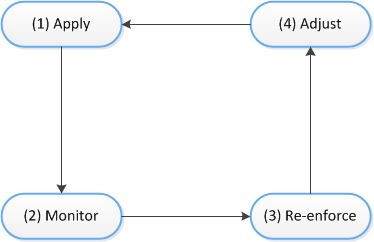
From the PTE Board, we can easily view Backlog, Active Sprints and Reports.

****

# RISKS

1. There might be cases where technical team is available for developing new features but PO team has not done with the requirements definition.
   1. Contingency plan: try to make enough lag for technical team i.e. PO teams concentrate to define requirements for critical features ahead of time e.g. at least one sprint ahead, so that technical team always have to-dos available for planning.
2. (?)

# WHAT’S NOW



# REFERENCES

N/A