Subscriptions in Jira

Projects > select project > issues and filters > Open issues > click advanced search > save as > Details > Edit permission > Details > New subscription > Recipients = Executive Management > click email this filter > subscribe

Visual dashboard alerts in Jira: (Set RAG for kanban / Sprint board)

Active Sprints > … > Broad setting > Column constraint = issue count > To Do (min = 5, max 8) , In Progress (min = 5, max = 10)

If item is more than max, will change the board color

Product roadmaps in JIRA

Projects > select project > Backlog > Versions > Create Version > drag items from product backlog into version >> create high level roadmaps

In gadget – JIRA road map > select project > select longer date

Two dimensional filter

Information radiators in JIRA (For present ppt like info to stakeholders and team members)

Dashboards > … > Create dashboard > add gadget > search “wallboard” > add “agile wallboard, days remaining in sprint, sprint burndown, & sprint health > Edit layout

… > view as wallboard

5. Scrum Events: Sprint Review

Sprint review is a collaborative informal event where the Scrum team and the external stakeholders collaborate to do two things.

1. They take stock of **where they stand** in terms of the progress towards building their product.
2. They discuss and decide on what they **want to do next**. This is an informal event and not a scripted product demo.

Inspect product increment and adapt product backlog

4 hours for 30 days

Product owner kicks off the event by projecting the sprint backlog and the sprint goal on a large electronic board. He provides additional updates on which of the items are ready for production and which of the items are not.

The stakeholders and the Scrum team continue to have informal conversations about the product features.

Scrum master is there to facilitate and she **makes sure the group is focused on the key goals of the event**.

Another stakeholder asks to provide clarifications on calendar of activities functionality.

The key thing to learn here is that the **sprint review is not a scripted product demo**. It does include an informal product demo but that review also involves stakeholders directly interacting with the product.

This is **followed by a conversation on what to do next**. The team decides to add a few product backlog items for a future discussion and de-prioritizes a few items once sprint backlog item with a major flaw was returned to the product backlog for a future sprint.

Close a sprint in JIRA:

Active sprints view > Complete sprint > select move to > complete > JIRA generates a sprint report about the closed sprint. The status report shows start date and end date of the sprint, list of completed issues, and issues that are not completed.

Confluence introduction:

Confluence is an excellent product for capturing documentation related to projects and is tightly integrated with JIRA.

Right side bar 9 square option > switch to “Confluence” > confirm > Continuous > go to confluence

Confluence has SPACES as folder of work area

Create a page in Confluence:

+ > page / blogs > create a page

Create meeting notes in Confluence

+ > meeting notes > create

Reports in JIRA:

Project > reports > velocity report >> show velocity of different sprint by story points

Project > reports > sprint report >> show info of completed or not

6. Scrum Events: Daily Retrospective

Sprint retrospective is the event to inspect and adapt everything other than the product increment and backlog. In other words, this is an event to discuss and create action items to become more efficient.

Answer the question: How can we become more efficient? & identify areas of improvement

All Scrum team members are required to attend this event.

In this event, the Scrum team discusses three things:

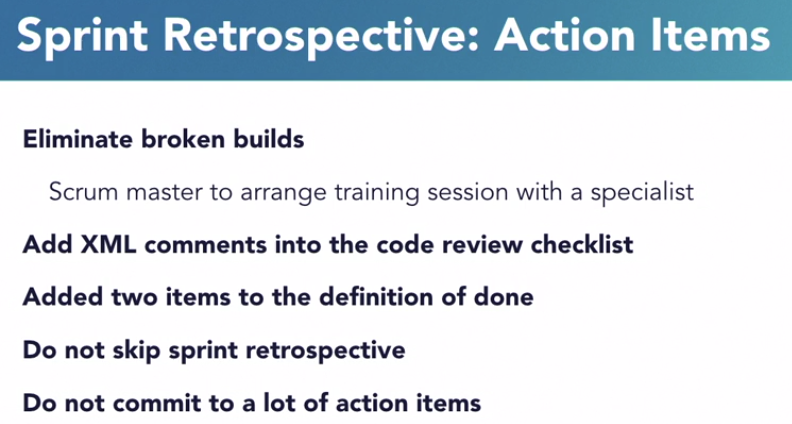
* what went well
* what could be done differently
* what do we commit to change?

3 hours

This is the event to discuss possible improvements and issues related to processes, tools, communication mechanism, team dynamics, et cetera.



The team selected a small subset of action items and those items were added to the product backlog and assigned to team members.



Advantages of keeping Sprint retrospective:

1. You will be surprised to identify areas of improvement even when you thought everything worked fine. There's always scope for improvement.
2. Sometimes the things that you did well may be the same things other teams are struggling with. If you document these items, those teams would get visibility to items and they could get your help.

Sprint retrospective items in Jira:

defined a new issue type called retrospective item and created a separate board in their Agile project to track those

JIRA > JIRA settings > Issues > Issue types > Add Issue type > Name = Retro item > Add

JIRA > JIRA settings > Issues > Issue types schemes > Edit > drag in it > save

Board > Create board > Scrum > from existing project > name = retro board > create board > … edit board setting > filter query > type = retro item

Sprint retrospective page in Confluence:

+ > Retrospective > next > create

Layout > Add section > add table >> use table to create tasks in JIRA

7. Product backlog refinement

Product backlog refinement:

The product owner and the development team are required attendees for backlog refinement

Unlike other events, the scrum team can conduct this event in one session or spread it over several sessions in multiple days. The guideline is that the **time spent on backlog refinement** **cannot exceed more than 10% of the development team's capacity**.

For example, for a two week or an 80 hour sprint, the development team members should not spend more than eight hours on product backlog refinement. E.g. 8 hours of 2 weeks (80 hrs) sprints.

The purpose of this event is to keep product backlog items detailed and ready for future sprints.

This event is used for the following activities:

* adding more details to product backlog items,
* clarify requirements,
* revisiting backlog item priorities,
* adding or removing backlog items to keep the product backlog in a cleaner state,
* dividing backlog items into smaller and more detailed items, and
* estimating backlog items.

PB refinement - clarifications:

For the user story on the ability to view training videos, the development team asked a bunch of questions. The product owner provided the clarifications.

PB refinement – split user stories:

The team split the online payment user story into three user stories. One each for credit card and online check and one for recurring payments.

PB refinement – add or update Backlog items:

The team continues to discuss the online payment user story and lower its priority. This is because most fitness club members can make a payment when they are at the fitness club.

They also added two new user stories. The team added estimates on a few detailed backlog items that are lined up for future sprints.

As we already discussed, pr**oduct backlog refinement facilitates smoother execution of sprint in general and sprint planning in particular**. It is a good diversion for the development team from their daily tasks and gives them a good overview of the bigger picture of the product roadmap.

Planning poker in Jira: to estimate backlog items

JIRA settings > add on > find new add ons > search planning poker (Spartez)

Project > Backlog > Agile Poker > Join standard session > go to session > Finish >

Project > Backlog > Agile Poker > Join standard session > Estimate

Project > Backlog > Agile Poker > Join standard session > view estimating and select the story points

If need, reset

If agree, save

Technical debt

A component needs refactoring, or rewrite, due to excessive dependencies. When you built the component, you had no idea that it will be so reusable, but now this component has so many dependencies that any change to this component would require retesting of the entire product suite.

Newer version of programming language and runtime are available. The product was built using an older version of the framework, and the new framework has many features deprecated or new features that should be used for optimal performance.

Quick short-term solution. A quick short-term solution was implemented due to competitor pressure and now, the solution needs to be rewritten for better maintenance and configurability.

Found a better way of implementing something. You did not know this new programming technique back when you implemented this feature, but now you can implement this feature with fewer lines of code.

Change in requirements. Due to change in law, you need to make a change to your product and that requires rewrite of several components.

And defects are always found in any product.

Technical debt is additional work caused by reasons discussed earlier. You can try to avoid technical debt but it is impossible to completely eliminate it. As you design and build software products, you accumulate technical debt.

So how should we address technical debt?

First of all, you should **make technical debt visible to technical and business stakeholders**. They need to understand why it is important to address technical debt.

Secondly, just like financial debt, you should **address technical debt as quickly as possible**. If you do not address financial debt, you accumulate additional burden in the shape of interest and penalties. In a similar manner, you should address technical debt quickly because features built on top of components with technical debt continue to be difficult to maintain. This also reduces transparency because the Scrum Team seems further along in their process than they actually are.

Lastly, you should **not spend an entire sprint just addressing technical debt**. You should always mix technical debt with business value.