

Scrum Development with JIRA & JIRA Agile Conclusion

Xavier Morera
www.xaviermorera.com
@xmorera



pluralsight 
hardcore developer training

JIRA Agile for Agile Scrum Development

- **Manage your Agile Scrum Project**
- **Using JIRA Agile as your ally**
- **Manage your project vs. letting your project manage you**
 - Firefighter mode – avoid like the plague
 - “Fail to prepare, prepare to fail”



Project

- Start by creating your project in JIRA
- Select type Agile Scrum



The Team

Committed

- Product Owner
- Scrum Master
- Team Members

Involved

- Stakeholders
- Customers
- Executive team



Requirements: User Stories and Epics

- Create your backlog



Sprint Planning Meeting

- Prioritize, estimate and create sub tasks (every sprint...)
- Sprint goal
- Commit

WANT
WANT
WANT
NEED

SET GOALS

- 1.
- 2.
- 3.



Daily Scrum

- Time boxed 15 minutes
- Communication & collaboration meeting
- Scrum master moderates and removes impediments
- Every attendee
 - What did you do yesterday?
 - What will you do today?
 - Is there any impediment on your way?



Sprint Retrospective

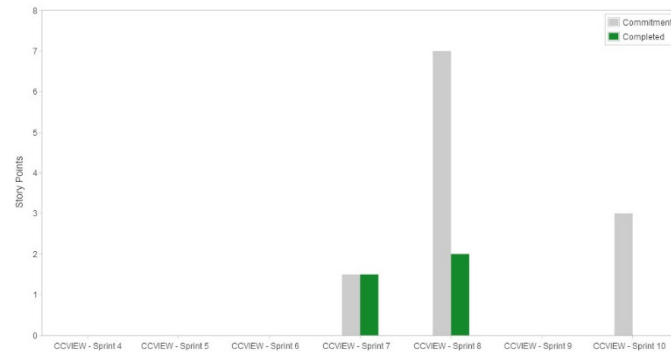
- Once Sprint is complete
- Suggestion: (What should we...)
 - Start doing?
 - Stop doing?
 - Continue doing?
- Prioritize, vote, act



Reports

- Help you understand state of affairs
- Multiple JIRA Agile reports
- JIRA general reports
- Build your own report

Velocity Chart

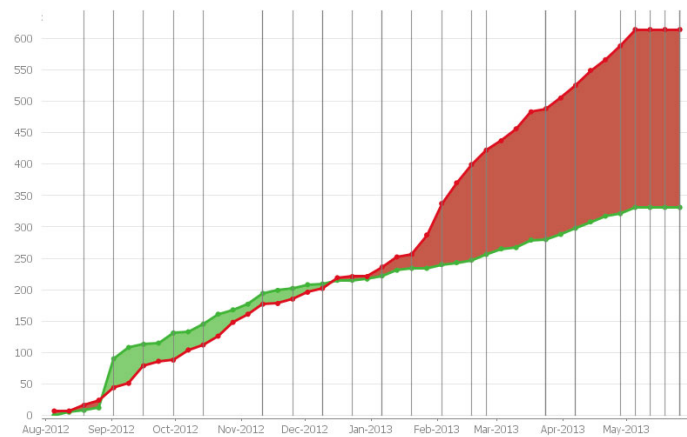


Sprint	Commitment	Completed
CCVIEW - Sprint 4	0	0
CCVIEW - Sprint 5	0	0
CCVIEW - Sprint 6	0	0
CCVIEW - Sprint 7	1.5	1.5
CCVIEW - Sprint 8	7	2
CCVIEW - Sprint 9	0	0
CCVIEW - Sprint 10	3	0

Project: CCView

Chart

This chart shows the number of issues **created** vs. the number of issues **resolved** in the last 300 days.



Sprint Review

- Dog & Pony
- Break a leg!
- Show what you've done



Multiple Teams – Same Project

- You may want to manage several teams in a single project
- Use Parallel Sprints
- Currently part of JIRA Agile labs
 - May disappear or change

JIRA Agile Labs

JIRA Agile Labs is a testing ground for new or experimental features that aren't quite ready for prime time. They may change or disappear at any time.

Features: ☐ **Parallel Sprints** - *Allows a new sprint to be started when one is already in progress.*

Issues Not Shown in Columns

- A status not mapped is an issue not shown
- This is an easy to make mistake

Add status

Add column

Unmapped Statuses

Drag statuses below to a column to map them

Statuses not containing issues

Reopened

No issues

☐ Set resolution

To Do

Drag to rearrange, or [delete](#)

No Min

No Max

To Do

No issues

☐ Set resolution

Open

9 issues

☐ Set resolution

In Progress

Drag to rearrange, or [delete](#)

No Min

2

In Progress

2 issues

☐ Set resolution

Done

Drag to rearrange, or [delete](#)

No Min

No Max

Done

No issues

☐ Set resolution

Closed

No issues

☒ Set resolution

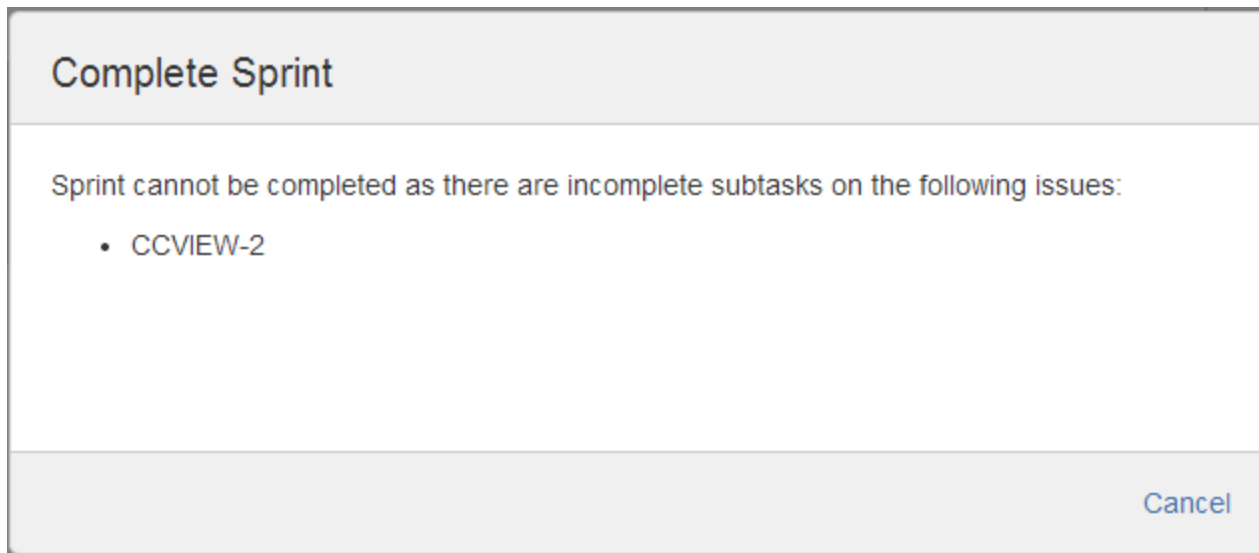
Resolved

1 issue

☒ Set resolution

Completing Sprint With Unfinished Subtask

- Cannot complete a sprint if a sub task is open but the parent task is resolved
- I usually convert subtask to task
 - Other people reopen parent and move everything to next sprint
- And the complete sprint





Sub-tasks Not Visible in Plan Mode

- They are not visible
- No way around it at this point


2 To Do

▼ Xavier MT 2 issues

 CCVIEW-2
↑ As a user I want to follow along with my peers 
1

 CCVIEW-27
↑ Sub task of 2 

16/Dec/13 4:58 PM • 30/Dec/13 4:58 PM

 ↑ CCVIEW-2 As a user I want to follow along with my peers

1

JIRA Agile Configuration


- JIRA Agile Configuration, Project Templates, Enabled Projects
 - Used for configuration of Classic Boards
 - Use the new ones

 The configuration options below only apply to Classic Planning Board, Classic Task Board, Classic Chart Board and Classic Released Board.

JIRA Agile Configuration

These general options can be changed for an individual project by the project administrator within the Project Configuration.

All changes will be saved automatically.

- Scheduling and Display ☐ **Scheduling Permission** - Use the Schedule Issues rather than the Resolve Issues permission for scheduling issues within JIRA Agile
- Options: ☒ **Card Creation** - Enable the creation of cards within JIRA Agile using the Card template. Disable this to force creation of issues via JIRA.
- ☒ **Time Aggregation** - Enable the aggregation of sub-task time tracking within the parent issue, denoted by a  icon.
- ☐ **Auto assign** - Enable the auto assign on card transition in the TaskBoard

Non-Working Days

Note: These dates will not be shown in your charts. All worklogs will be associated to the previous working day. When editing the non-working days the timezone in your [user profile](#) must match the server timezone.

Non-Working Days: 

JIRA Agile Listeners and Services

You can register directly from here the services and listeners of JIRA Agile.

Epic Migration


- Migrate old Epics (Greenhopper) to new Epics
- Once you migrate, old epics are not synchronized

Classic Epic Migration

Migrates Classic epic links for use in new boards.

Start Classic Epic Migration

Classic Epic Migration - Select Projects

 This action will migrate existing epic links. It does not synchronise the Classic epic links over time. This action can be run again at a later time if necessary. Existing epic links on the new board will not be affected.

Projects*

Select projects for epic migration

[Cancel](#) [Next](#)

Keyboard Shortcuts

- Be efficient! Learn the shortcuts

Keyboard Shortcuts

Global Shortcuts

Go to Dashboard: **g** then **d**

Next Item: **j**

Browse to a Project: **g** then **p**

Go to Agile: **g** then **a**

Previous Item: **k**

Find Issues: **g** then **i**

Administration Quick Search: **g** then **g**

Go to Tempo: **g** then **t**

Quick Search: **/**

Quick Operations: **.**

Create an Issue: **c**

Log work with Tempo: **t**

Plan Time with Tempo: **p**

Open shortcut help: **?**

Navigating Issues

View selected Issue: **o** or **Enter**

Next Issue: **j**

Previous Issue: **k**

Dock/Undock the Filters Panel: **[**

Next Activity: **n**

Previous Activity: **p**

Focus search field: **f**

Search for issues: **u**

Switch filter view: **t**

Assign To Me: **i**

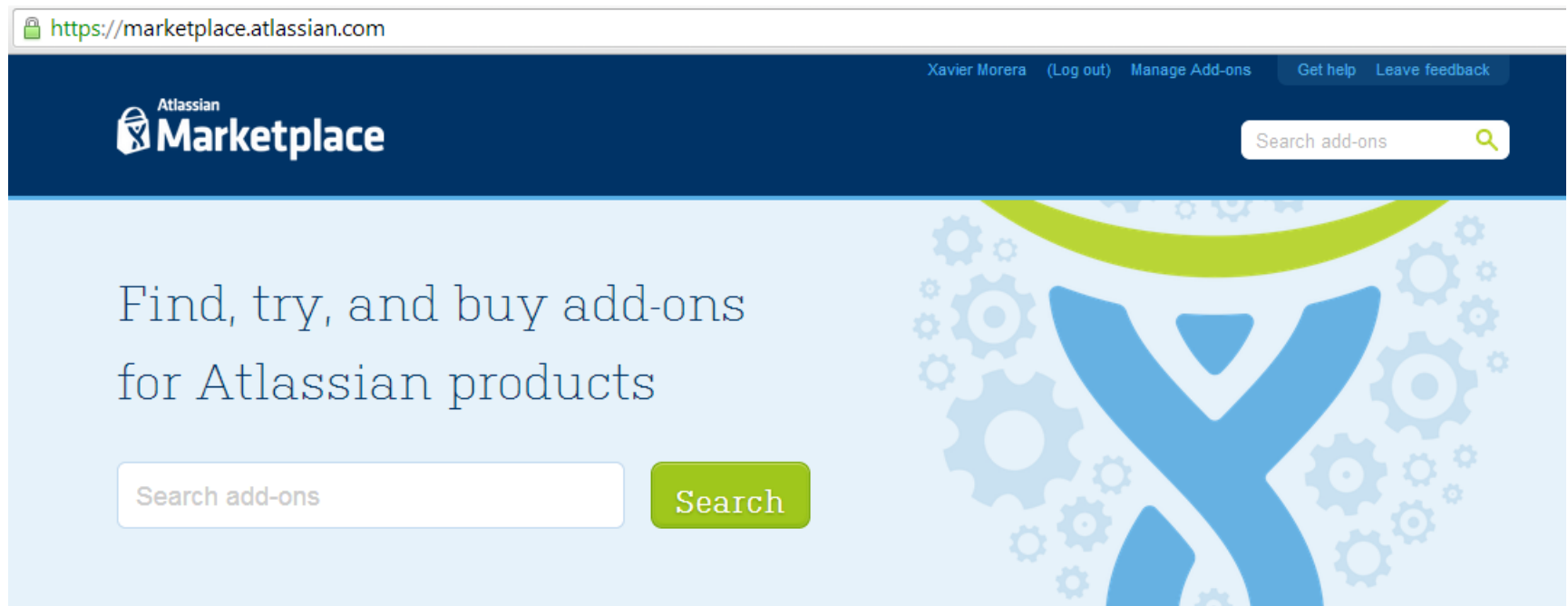
Detail View Order by: **z**

Keyboard Shortcuts are **enabled**. [Disable Keyboard Shortcuts](#)

[Close](#)

Marketplace

- Don't reinvent the wheel
- Somebody might have already created what you need



Remote API

- If not available or you want to build it yourself?
- Remote API → Suggest REST
 - Deprecated options also available

JIRA Development / JIRA Plugin Development

⚙ Tools ▾

Remote API Reference

JIRA Remote API	Status
REST	Recommended and supported. REST is the recommended and supported remote API in JIRA 5.0 and later. This is where Atlassian will focus its attention from this point onwards, adding functionality and fixing bugs.
XML-RPC	Deprecated in JIRA 6.0. This means that we will not add new features nor fix bugs related to the XML-RPC API during the JIRA 6.x lifecycle. Read the deprecation announcement for more information.
JSON-RPC	Deprecated in JIRA 6.0. This means that we will not add new features nor fix bugs related to the JSON-RPC API during the JIRA 6.x lifecycle. Read the deprecation announcement for more information. Atlassian added JSON-RPC capabilities to JIRA to help you move from the SOAP API to a mechanism that uses more JSON and JavaScript, on the way towards the REST APIs. We recommend that you move to REST as soon as possible. For information on the deprecated SOAP APIs, see Creating a JIRA SOAP Client .

Plugin Framework

- Want to build your own add-on?
- You can also post it to the Marketplace!

[Documentation](#) / [Home](#) / [Atlassian Platform Common Components](#)

 [Tools](#) ▾

Plugin Framework

Getting Started

Get started by [setting up your Atlassian plugin development environment](#).

With Atlassian's plugin development platform, you can create plugins that extend the functionality of Atlassian applications such as [JIRA](#), [Confluence](#) and [others](#).

The platform consists of a plugin framework and set of components that provide useful tools to plugin developers. To see how the pieces fit together, take a look at the [overview of the platform](#). To get started writing plugins, download and set up the Atlassian Plugin SDK. Then follow the plugin development guidelines for your specific application, as listed [below](#).

JIRA Source Code

- Commercial users receive access to JIRA's source code
- Change, customize and redeploy

 **Atlassian Developers**

JIRA Development

- > JIRA Plugin Guide
- > Tutorials
- > Development Resources
- > Remote API Reference
- > Java API Reference
- > JIRA Agile Development
- > FAQ
 - JIRA Custom Installer Guide
- > JIRA Mobile Connect

Learn how to write Gadgets to expose or consume content in Atlassian applications.

How JIRA works

See what is really going on inside JIRA. Some of this information is useful to plugin developers. Other pieces are more relevant to the JIRA development team, but we have published them here in the spirit of open documentation.

JIRA source code

Commercial users receive access to JIRA's source code. Learn how to build this source code back into a deployable JIRA application.

The End

- Thanks for your time!
- Happy scrumming
- May your project be successful!

THANK YOU

