

Recipes for: project  
managers, agile teams,  
and anyone in a hurry

*the quick and easy*  
**Jira reporting  
cookbook**





Preface from the Old Street chefs..... 3

A note about ingredient choices: Reports or gadgets?..... 5

## Recipes

1. One-team dashboard for the daily standup..... 7

2. Dashboard for retrospective meetings..... 8

3. Dashboard for reporting on programs..... 9

4. Jira dashboard reporting in Confluence..... 11

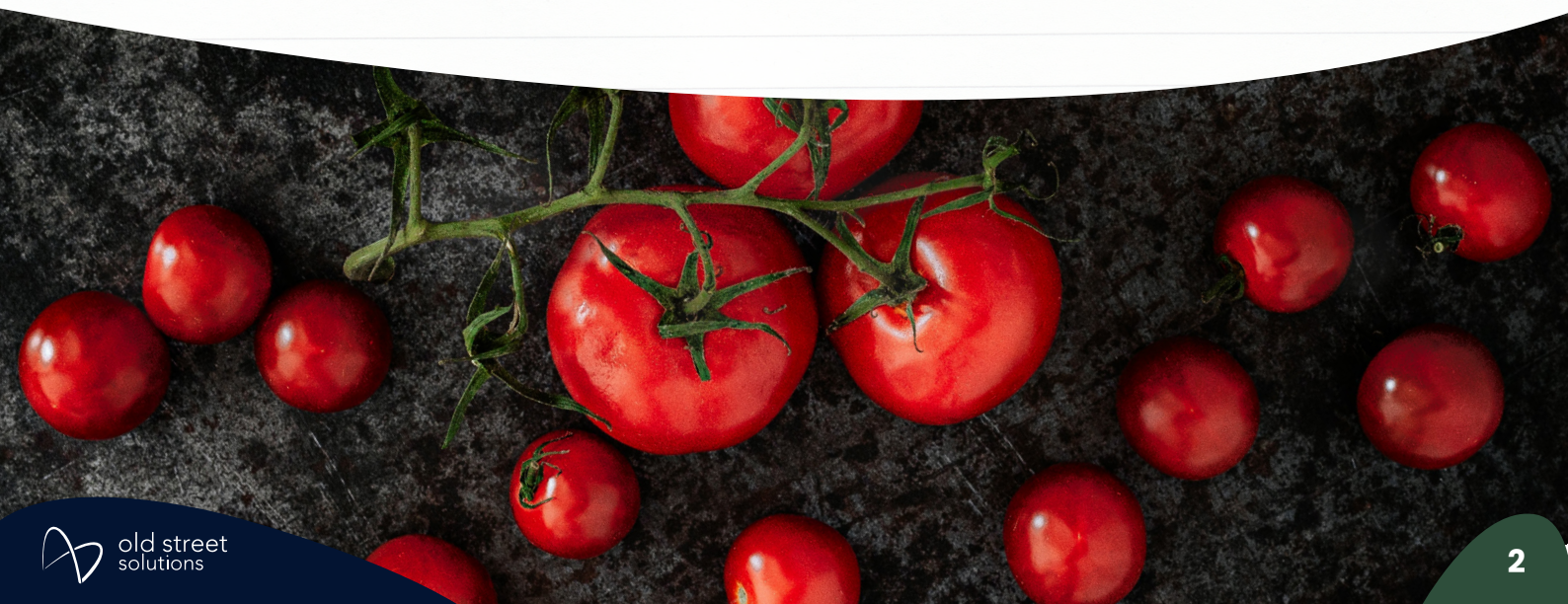
5. Customer satisfaction dashboard for JSm..... 13

6. Dashboards that don't just count issues..... 16

7. JQL-infused dashboard for Scrum Masters..... 18

8. Advanced-level dashboard for project  
management organizations..... 21

Who's hungry?..... 24





# Preface from the **Old Street chefs**

Jira reporting is versatile, and can be as technically challenging as cooking a seven-course gourmet dinner. That's great if it's what you're after, but sometimes all you want is a grilled cheese sandwich. With this cookbook, you'll know what you're getting yourself into before you start, and have the right tools and knowledge to whip up whatever takes your fancy!

You can't make a great cake with too few ingredients. But too many ingredients is a recipe for disaster too. Adopting super-powerful big data business intelligence platforms may give your company the most comprehensive reporting options. But they come packed with capabilities you probably don't need, and are so complex that most users won't have a clue where to start.

*It's like going  
from baking this...*



*...to baking this*





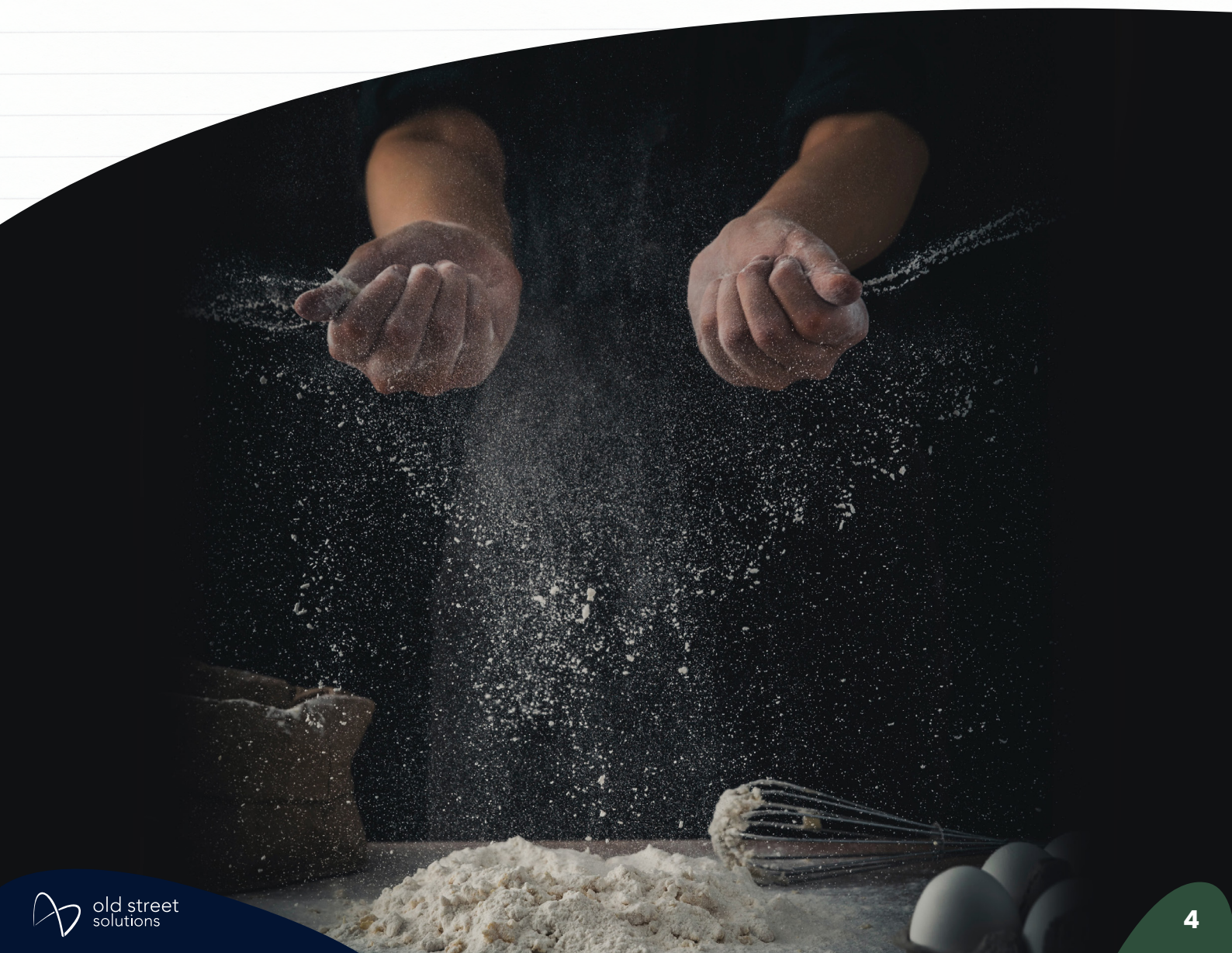
It means that overwhelmed Jira newbies and business users may avoid custom reporting altogether, sticking to the native capabilities and never really creating the reports they want to (i.e. the ones that'll really make a difference to their team's performance). The knock-on effect? A shortage of knowledge, leading to a loss of agile-shaped value. One of the core principles of the Agile Manifesto is that the team should regularly reflect on how to become more effective, before adjusting its behavior accordingly. Can't do that without clear reports, can you?

An equally problematic alternative is that organizations hire a full-time dedicated expert to code its most complicated, fancy-schmancy reports. This creates a 'reporting overlord' in the organization – someone who hoards all the knowledge, visibility, and control of the company's data. Not only does this lead to knowledge and time bottlenecks in day-to-day working, it's also more expensive (because the programming expertise needs to be paid for) and higher risk (because the expert could leave, plunging you into a reporting blackout).

None of that's good, frankly, which is why we've created this cookbook. Inside, you'll find quick and easy recipes for a whole bunch of delectable Jira reporting concoctions to improve team efficiency and productivity.

For slightly more experienced Jira chefs, there's a recipe for a more targeted dashboard infused with Jira Query Language (JQL) flavors. For Jira masterchefs, there's a recipe for an advanced-level business intelligence dashboard that will make your Jira snap, crackle, and pop.

*Now let's get on with making jira delicious for all!*







## A note about ingredient choices:

# Reports or gadgets?

Before we get into the recipes, let's talk about the two main ingredients people use to make reports in Jira:

- Reports
- Gadgets

A report is a chart, graph, or diagram generated using your Jira data, which you access and configure by going into an individual project and selecting from the options available. In Jira Software, you'll then need to navigate to a board. There are various reports built into Jira, such as the Sprint Report, Velocity Chart, and Cumulative Flow Diagram.

Project reports are created and displayed one by one and present users with a point-in-time view of the data, i.e. the data is fixed at the time you run the report. You'd need to run the report again to update it.

A dashboard gadget is also a chart, graph, or diagram generated using your Jira data, but which you access and configure directly on a Jira dashboard. Each gadget appears as a window or block on your dashboard screen. As with reports, there are various gadgets built into Jira, such as Two Dimensional Filter Statistics, Sprint Health, and Filter Results.

A Jira dashboard displays multiple gadgets and each one is dynamic rather than a snapshot. In other words it refreshes automatically every time you load the dashboard, or you can configure it to refresh at set intervals, thereby offering users a near-real-time view of their data.

There are some charts that come in both project report and dashboard gadget form, such as the Created versus Resolved Chart, Burndown Chart, and Average Age Chart. There are also some charts that are gadget-only and some that are report-only. For example, the Velocity Chart is only a report and Sprint Health is only a gadget.

Generally, project managers will use project reports when they want their teams and stakeholders to engage with a specific set of data. For example, they might want to highlight bottlenecks using a Cumulative Flow Diagram to show how long work items are in their various statuses. They'll also use reports if the report they need isn't available as a gadget on a dashboard.

Dashboards are capable of displaying multiple charts and graphs at once, in order to give teams and stakeholders a broader picture of what's going on. For example, you can create a dashboard to cover multiple projects and boards, whereas a report is confined to one board in one project only.



That said, you can use a dashboard for more focused reporting as well. You can, for example, maximize a gadget so that it appears full screen. Or you could, if you wanted to, build a dashboard with only one gadget displayed. It's entirely possible to use gadgets to get teams to home in on something specific, as you would with a report.

You can also bypass the problem of some charts only coming in project report form in Jira Service Management (JSM). JSM ships with a frickin' awesome gadget called the Service Project Report gadget, enabling you to embed project-report-only charts into your dashboard, such as your SLA reports. More on this later.

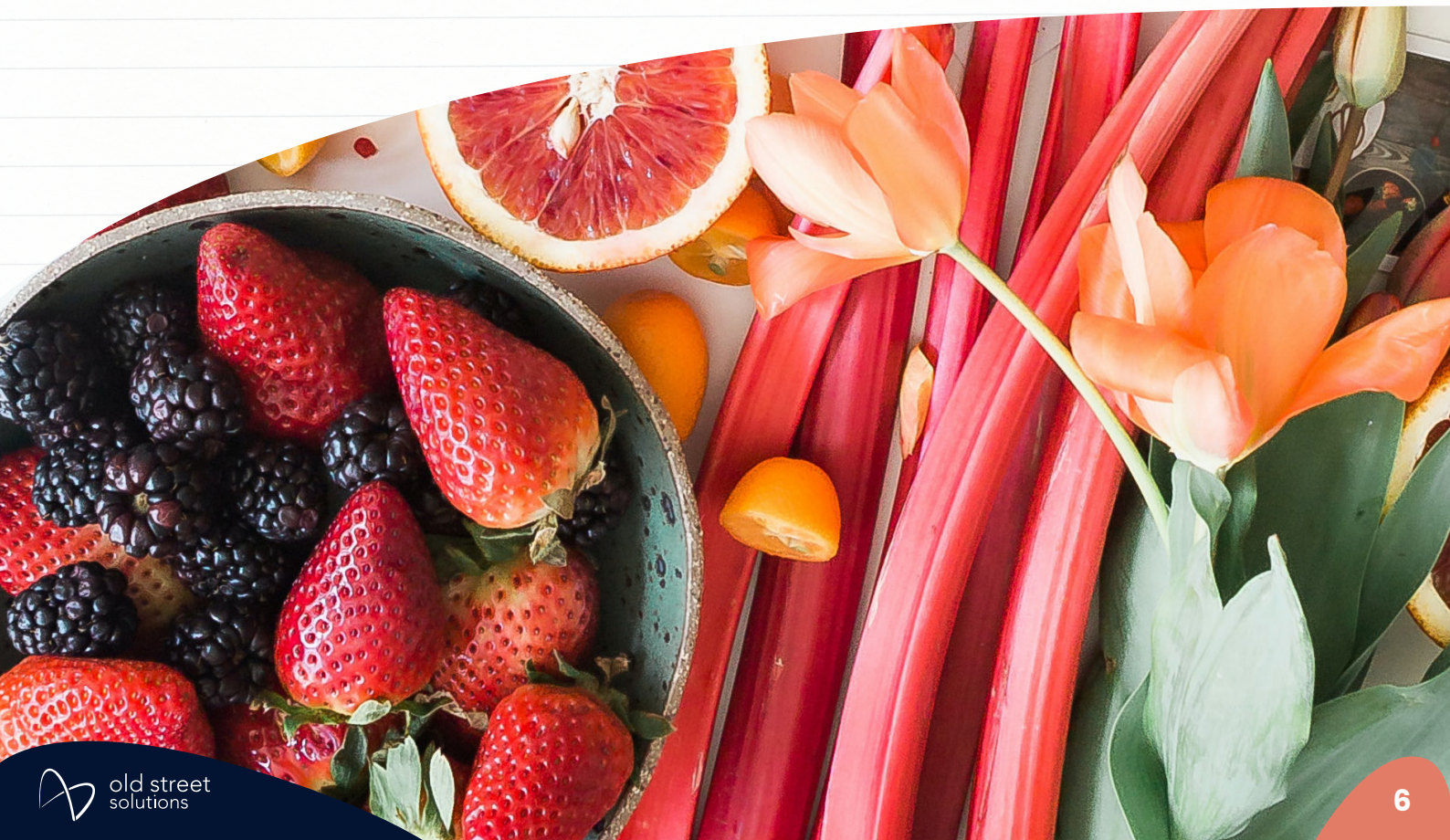
## So which is the better way to report in Jira?

If you think it sounds like we're suggesting that dashboard reporting is better, then... you'd be right. We are and it is. Gadgets are just more versatile. You don't have to go looking for them because they're all right there in the dashboard, and you don't need to open them one by one on separate screens. You can display as many or as few gadgets as you want. And they will automatically refresh with new data as often as you want them to.

Plus, although the learning curve with native Jira is as flat as a botched soufflé, a botched soufflé isn't all that inspiring to look at, or eat. (Yeah, we know you don't eat reports; go with it.) With a Jira dashboard, you can get add-ons like Custom Charts for Jira and swap out some of the botched soufflés – ahem, built-in gadgets – for more dynamic and nicer-looking customizable versions. That's another reason why you should maximize your use of Jira dashboards and only use reports when you have to.

To be honest, we think the best use of project reports is when a gadget version isn't available. For example, you can't stick a Velocity Chart or a Cumulative Flow Diagram on a dashboard, and there will be times when your teams need to view those reports.

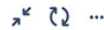
Now that we've established that the tastiest reports in Jira can be found on the Jira dashboard, let's get started with some simple and succulent dashboard recipes.







## Sprint Health Gadget



## SP Sprint 4 - SP board

Overall sprint progress (Story Points)

1 day left



50 %

Time elapsed

82 %

Work complete

21 %

Scope change

0

Highest

0

Flagged

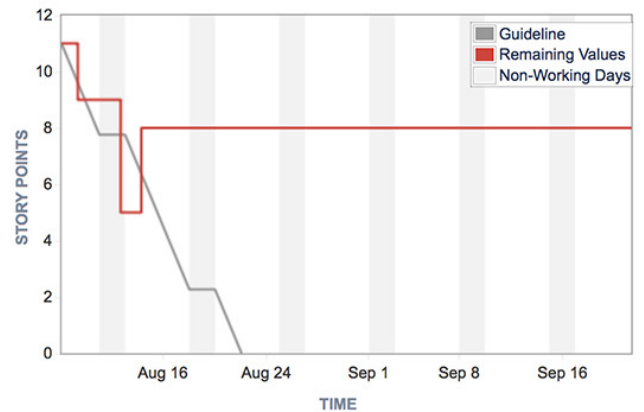
## Filter Results: Flagged Development Tickets



T	Key	Summary	Status	Assignee	Epic Link
	CC-956	Add a 'Save As' button	TO DO	Rebecca Lopez	UX Enhancement
	CC-936	Add Auto-refresh to configuration options	TO DO	Mary Smith	Backend Dev

1-6 of 6

## Sprint Burndown Gadget



You will need:

- Jira's Sprint Health Gadget
- Jira's Sprint Burndown Gadget
- Jira's Filter Results Gadget
- Standard Jira Fields



Difficulty level

Easy



Serves

Agile Teams

## Recipe One

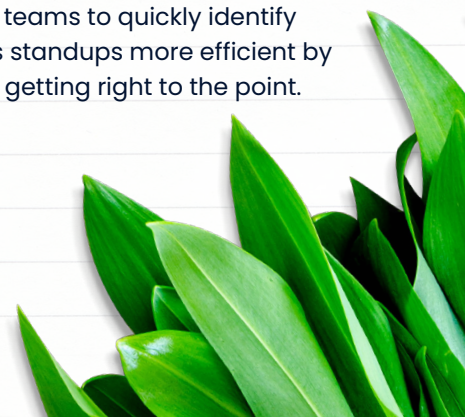
## One-team dashboard for the daily standup

## Method

Combine these three native Jira gadgets and you'll end up with a truly delicious dashboard for triggering conversations about how the sprint is going at the daily standup.

Sprint Burndown and Sprint Health let you see the progress of the team over the course of the sprint. Sprint Burndown looks at the history of the sprint and tells you if your team is on track to deliver based on how close you are to the completion guideline (the grey line). Sprint Health gives you clear percentages beneath a color-coded bar chart of "To Do", "In Progress", and "Done" items. Sprint Health also lets you know what your scope change is, together with any blockers and flagged issues. These are a great heads up for your Scrum Master when it comes to roadblock removal.

Add to these the Filter Results gadget showing a list of specific blockers and their assignees. Although it's less visual than the other two charts, it can be an extremely valuable tool for daily standups, enabling teams to quickly identify issues that are holding up people's work. This makes standups more efficient by putting the most important information upfront and getting right to the point.





## Jira Road Map: Next 365 Days (Until 09/Dec/22)

Scrum Project :Phase 2 Milestone	30/Jul/21	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
		2 of 3 issues resolved.
Scrum Project :Phase 3 Milestone	31/Aug/21	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
		3 of 3 issues resolved.

## Filter counts: Scrum Project

6

Status - To Do

4

Status - In Progress

8

Status - Done

## Created vs. Resolved Chart: Scrum Project



Issues in the last 180 days (grouped weekly)

View in Issue navigator

○ Created issues (6)

○ Resolved issues (5)



You will need:

- Jira's Road Map Gadget
- Jira's Created vs Resolved Gadget
- Jira's Filter Count Gadget
- Standard Jira Fields



Difficulty level  
Easy



Serves  
Agile Teams

## Recipe Two

### Dashboard for retrospective meetings

#### Method

A dashboard made up of three native gadgets designed to look at issue status, scope creep, and progress on upcoming versions serves as an excellent backdrop for a retrospective.

The Road Map gadget shows a list of versions due for release within a specified period of time, together with color-coded bars showing the number of issues resolved and unresolved. This instant visual summary of progress made towards completing those versions lets teams check whether they've been focusing on the right balance of issues.

The Created vs Resolved gadget shows a difference chart of issues created versus resolved in a given period. If you set that given period as the sprint you've just completed, then any issues created during it can be considered scope creep. Identifying and understanding scope creep on a sprint can help you avoid it in the next one and potentially understand why the agreed-upon sprint scope was not completed as expected.

Finally, the Filter Count gadget shows counts of issues based on certain filters. Use "status" as the filter to see how many issues are in each status, so your team can talk through the issues that weren't completed in the previous sprint. You can then discuss whether those issues should be rolled forward to the next sprint, or returned to the backlog.





## Program Status Dashboard



Add gadget

Change layout

Done



### Jira Road Map: Next 30 Days (Until 08/Jan/22)



Scrum Project :Phase 2 Milestone	30/Jul/21	2 of 3 issues resolved.
Scrum Project :Phase 3 Milestone	31/Aug/21	3 of 3 issues resolved.

### Custom Charts



### Custom Charts



### Program Status: RAG



#	RAG	Issues	%
1	Red	8	44.4%
2	Amber	6	33.3%
3	Green	4	22.2%
Total		18	100%

**RED** - A forecast overspend of more than 10%

**AMBER** - A forecast overspend of more than 5%

**GREEN** - Forecast expenditure is on budget



You will need:

- Jira's Road Map Gadget
- Custom Charts' RAG Status Pie Chart
- Custom Charts' Due Date Bar Chart
- Standard + Custom Fields

## Recipe Three

### Dashboard for reporting on programs

#### Method

This dashboard is designed to give you an overview of the status of multiple related projects, like in a program. Sometimes these projects are dependent on issues from other projects, or are part of a larger portfolio of work that is tracked as a single unit. For program or project managers, this view of cross-project status and progress can illuminate the bigger picture.

First, add Jira's Road Map gadget to your dashboard. This shows a list of versions due for release and the number of issues resolved/unresolved in each version. As a lot of programs use versions in Jira to represent project phases or work toward major milestones, the Road Map offers a nice visual summary of how the program as a whole is coming along.

Then you want to add a RAG Pie Chart to your dashboard. RAG stands for red, amber, and green and is a great way of demonstrating program health with an instantly understandable visual. For a program, you're likely to use RAG to see whether you're on a budget, on schedule, or whether you have enough people to do the work.





## Program Status Dashboard



Add gadget

Change layout

Done

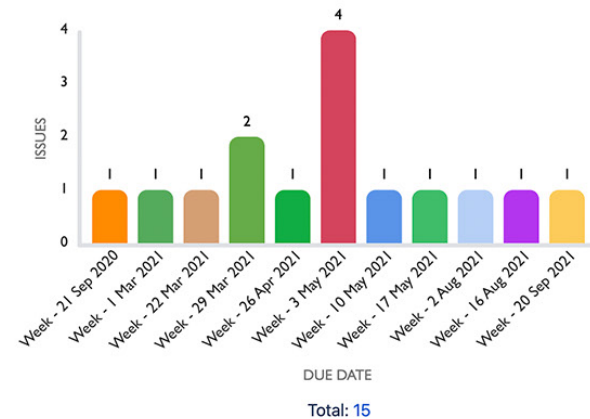


### Jira Road Map: Next 30 Days (Until 08/Jan/22)



Scrum Project :Phase 2 Milestone	30/Jul/21	2 of 3 issues resolved.
Scrum Project :Phase 3 Milestone	31/Aug/21	3 of 3 issues resolved.

### Custom Charts



### Custom Charts



### Program Status: RAG



#	RAG	Issues	%
1	Red	8	44.4%
2	Amber	6	33.3%
3	Green	4	22.2%
Total		18	100%

**RED** - A forecast overspend of more than 10%

**AMBER** - A forecast overspend of more than 5%

**GREEN** - Forecast expenditure is on budget

Continued...



Difficulty level  
Easy



Serves  
Program Managers,  
Project Managers

## Recipe Three

### Dashboard for reporting on programs - continued

#### Method

To track this in Jira, simply create a RAG custom field with red, amber, and green dropdown options, and reference the field in your dashboard gadgets. Unfortunately, making a RAG chart in native Jira is like making a chocolate cake without cocoa. It's not actually a RAG chart at all because RAG coloring isn't available. So, to customize colors in your charts, install a marketplace app like [Custom Charts for Jira](#). Add the Custom Charts gadget to your dashboard and now you can create a pie chart with RAG colors.

Finally, add a Due Date report. This can show issues due soon across all of the projects in the program. You can do this in native Jira with a Filter Results gadget. However, Filter Results will only display the data in table/list form. Since you've already installed Custom Charts for the purposes of creating a proper RAG chart, you may as well get your money's worth from the app. Create a bar chart based on Due Date for a much clearer and more memorable summary of issues approaching their deadlines.





## Beautiful Confluence Reports

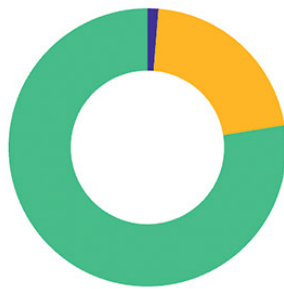


Created by Morgan Folsom

Last updated: just a moment ago • 0 min read • Analytics

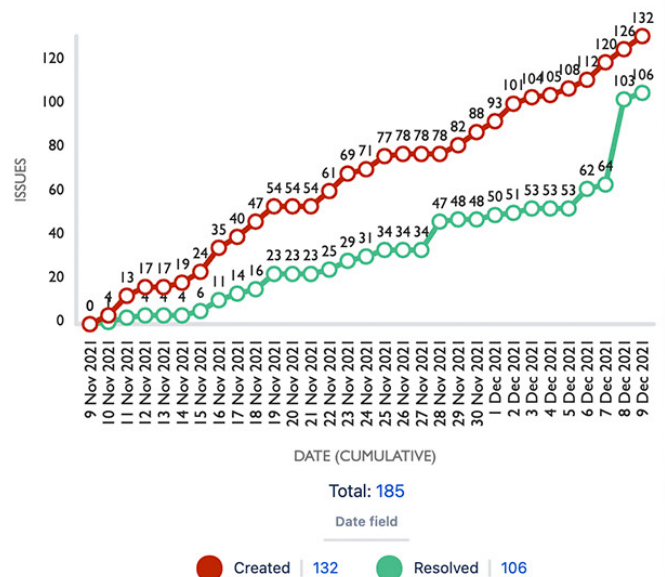
CC-955: Dark features page enhancements IN PROGRESS

Status Categories



Status Category	Issues	%
To Do	25	1.3%
In Progress	415	21.2%
Done	1521	77.6%
Total	1961	100%

Created vs Resolved - Past 30 Days



You will need:

- Jira Issues Macro
- Jira Charts Macro **OR** Custom Jira Charts for Confluence Macro
- Standard Jira Fields

**Optional additions:**

- Screenshots
- Attached spreadsheets

## Recipe Four

### Jira dashboard reporting in Confluence

#### Method

Many teams, particularly business teams and managers, spend way more of their time in Confluence than Jira. But they still need to see how projects in Jira are progressing. As the integration between these two platforms is so strong, options for visualizing Jira issue data in Confluence are plentiful.

The most popular option is the Jira issues macro. This macro allows you to embed a single Jira issue into a Confluence page, create new issues from within Confluence, or display a list of issues as filter results. The filter results view is a crowd favorite and particularly useful for meetings or agile ceremonies like retrospectives, or when you're looking to pull together a project status report. You're able to customize which columns are displayed and which issues are pulled in, just like the Jira issue search. Because this macro pulls issue data directly from Jira, no manual updating is required.

Add to this the Jira charts macro, which offers a few limited reporting options. These charts generally operate in the same way as their Jira dashboard counterparts, enabling you to see basic pie charts, two-dimensional filter statistics, and created versus resolved issues on a Confluence page. What's nice about this macro is that it pulls in live data capable of being dynamically interacted with and updated in real time, keeping your Confluence pages relevant for much longer.

The problem is how utilitarian they are. You have a couple of options for selecting what information you want to pull into the charts, otherwise (just like Jira dashboard gadgets) you can't customize how they appear, e.g. ordering, hiding specific data points, adjusting colors etc.





## Beautiful Confluence Reports

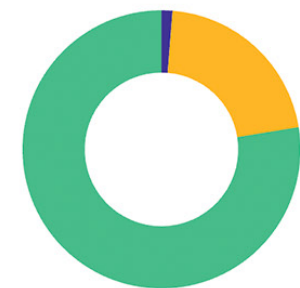


Created by Morgan Folsom

Last updated: just a moment ago • 0 min read • Analytics

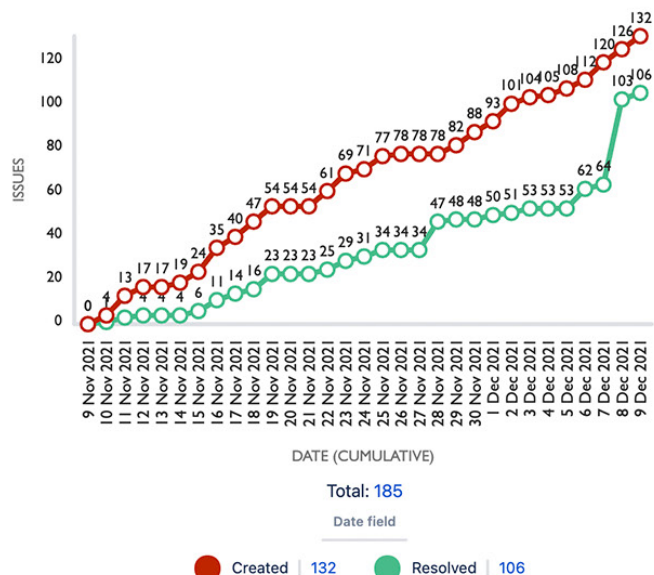
CC-955: Dark features page enhancements IN PROGRESS

Status Categories



Status Category	Issues	%
To Do	25	1.3%
In Progress	415	21.2%
Done	1521	77.6%
Total	1961	100%

Created vs Resolved - Past 30 Days



Continued...



Difficulty level  
Easy



Serves  
Business Teams,  
Leadership,  
Everyone

## Recipe Four

### Jira dashboard reporting in Confluence - continued

#### Method

Instead, you might want to use [Custom Jira Charts for Confluence](#). Insert this macro into your Confluence page and you'll be able to build a whole range of stunning, fit-for-purpose reports directly in Confluence, including pie charts, bar charts, stacked bar charts, tile charts, funnel charts, and more. And you'll be able to customize colors, hide/show data as you need, and rename variables to make the information more user-friendly.

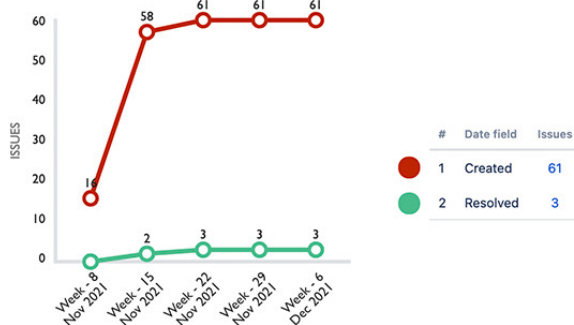
Custom Charts also has some specific features to enhance your Confluence reporting. The first is Simple Search. This is an additional macro that comes with Custom Charts and allows users to dynamically filter multiple charts at once on the page, drilling down into the data however they like. The second is User Impersonation. This feature, which must be turned on by your Confluence administrator, enables you to have your charts loaded as a specific user. This means that if you have execs or customers viewing Confluence pages but who don't have access to your Jira, you'll be able to load charts so that they can see the data.

You of course have the option of adding images and screenshots, and attaching spreadsheets for further information. Utilizing several options at once gives your team the most scope for reporting in Confluence effectively, and the benefit of Confluence is that it harbors a ton of flexibility in how data is displayed.





## Custom Charts



## SLA met vs breached

13 Met  
4 Breached



## Custom Charts



#	Organizations	Issues	%
1	Company A	28	96.6%
2	Company B	1	3.4%
Total		29	100%

## Custom Charts

Jerry  
5

Tom  
3.4

## Custom Charts



#	Assignee	Issues	%
1	Morgan	21	70%
2	Tom	6	20%
3	Chris	3	10%
Total		30	100%



You will need:

- Jira's Service Project Report Gadget
- Custom Charts for Jira's Customer Satisfaction Report
- Custom Charts for Jira's Created vs Resolved Chart
- Custom Charts for Jira's Workload Report
- Custom Charts for Jira's Organizations Funnel Chart
- Standard Jira Fields

## Recipe Five

### Customer satisfaction dashboard for JSM

#### Method

Understanding what customers are asking for, whether agents are keeping on top of what they're asking for, and whether customers are happy with the resolutions you're giving them, is all essential knowledge for a Jira Service Management (JSM) support team.

To build a really palatable dashboard for measuring these metrics, you first need to produce some project reports. As we pointed out earlier, most project reports can't be pulled onto a dashboard; a gadget version has to exist if you want it on a dashboard. However, there's an absolutely scrumptious gadget that lets you do this with a whole bunch of Service Level Agreement (SLA) reports. It's called the Service Project Report gadget.

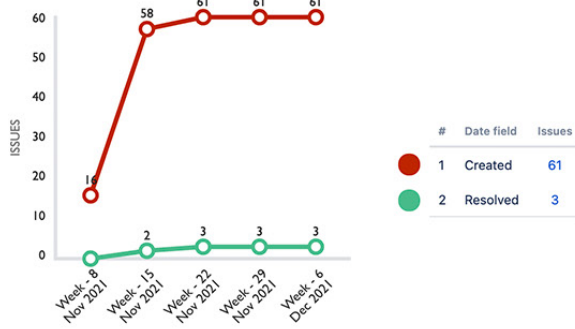
So, start by generating the SLA reports that will show whether you are meeting the expectations set out in your SLAs, such as SLA Met vs Breached (which compares the number of requests that have met or breached an SLA goal) and SLA Success Rate (which shows how your team is tracking toward their SLA goals). You can also create your own SLA reports in the JSM Custom Reports tab.

Then, add the Service Project Report gadget to your dashboard and pull in each of those reports so that they appear as separate gadgets.



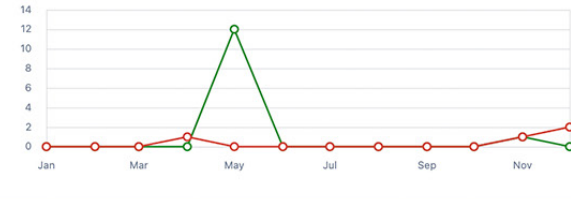


## Custom Charts



## SLA met vs breached

13 Met  
4 Breached



## Custom Charts



## Custom Charts

Jerry  
5

Tom  
3.4

## Custom Charts



Continued...



Difficulty level  
Easy



Serves  
Support Teams,  
Service Managers

## Recipe Five

### Customer satisfaction dashboard for JSM - continued

#### Method

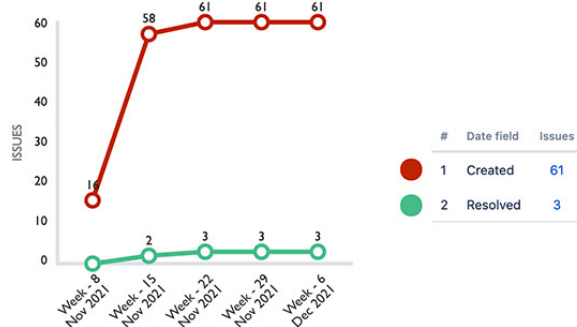
Next, add a Custom Charts for Jira gadget to your dashboard and create a Customer Satisfaction (CSAT) chart by calculating the Satisfaction field. Although there is a CSAT project report with native Jira, it doesn't come in gadget form, so you can't display it on a dashboard (and no, it can't be pulled over with the Service Project Report gadget either, in case you're wondering). Also, the native CSAT report comes in a simple table format that only allows you to filter by dates that the rating was received to view a list and average of the results for a single project. Not only is your CSAT data viewable on a dashboard with Custom Charts, you can also generate many different types of charts to visualize this data. These can span multiple projects and you can filter down further based on assignee or any other custom field.

As a matter of fact, you'll find that many of the native gadgets are highly restrictive in terms of how you can visualize the data. They'll just about get the basic job done, but the lack of customization may prevent you from getting your message across in the way you intend. With Custom Charts you can replace most of your native gadgets with Custom Charts versions. For example, add a Custom Charts Created vs Resolved chart and you'll be able to change colors, labels, filter the data, and add a description, giving your team a more visual, more focused view of whether you're keeping up with all of the work that is coming in.





## Custom Charts



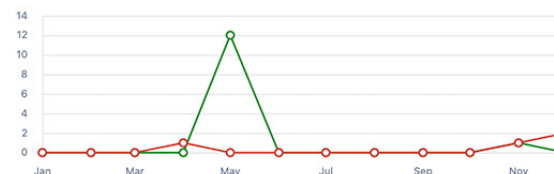
## Custom Charts



#	Organizations	Issues	%
1	Company A	28	96.6%
2	Company B	1	3.4%
Total		29	100%

## SLA met vs breached

13 Met  
4 Breached



## Custom Charts

Jerry  
5

Tom  
3.4

## Custom Charts



#	Assignee	Issues	%
1	Morgan	21	70%
2	Tom	6	20%
3	Chris	3	10%
Total		30	100%

## Recipe Five

### Customer satisfaction dashboard for JSM - continued

#### Method

The Workload Pie Chart helps you understand how the work among your service desk agents is distributed, so that you can provide support to those who may have too much on their plate. This is another native gadget that can be swapped out for a Custom Charts version if you so wish. If you've already added Custom Charts in order to get CSAT information on your dashboard, you might as well. While the native version is fine if you're happy reporting on the data in a fairly raw state, the Custom Charts version lets you show/hide specific assignees, chart by original/remaining estimates or time spent, and customize colors.

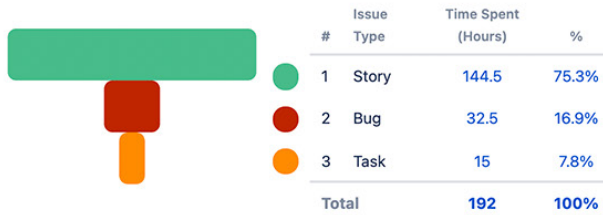
Finally, there's actually no way of charting by Organization natively on a Jira dashboard, so if you want to see where requests are coming from across your customer base, you'll need to produce a Custom Charts for Jira Organizations chart. We recommend a funnel chart, but if you prefer you can choose a different kind of chart to see the data in another format.

An Organizations chart can also be used to drill into the issues and identify trends. For example, it would be useful to know if one organization submits lots of feature requests while another submits lots of bugs. This information can be used to spark conversations about whether customers are happy, and ensure that you can make them so with future work.



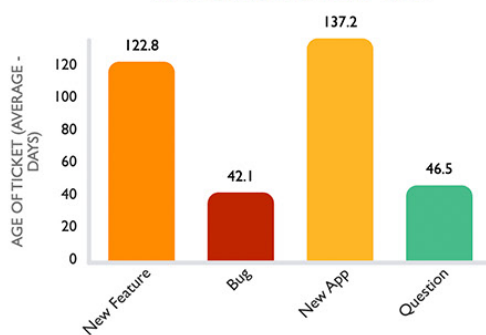


## Custom Charts

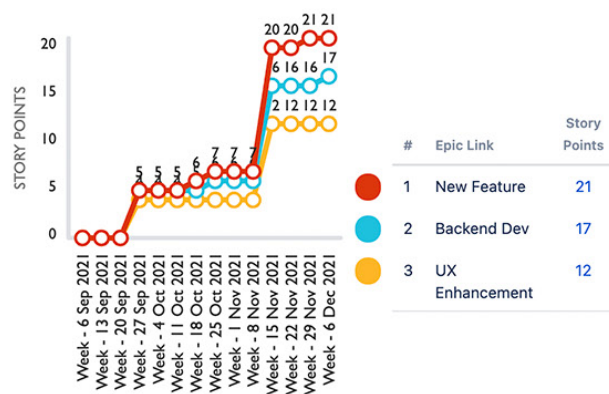


## Custom Charts

Average Age by Issue Type



## Custom Charts



## Custom Charts

Story Points by Assignee

Assignee	Sprint 1	Sprint 2	Sprint 3	Total
Courtney	7	8	14	27
Rebecca	5	-	-	5
Total (Story Points)	12	8	14	32



You will need:

- Custom Charts for Jira **OR** Custom Jira Charts for Confluence
- Custom Charts' Story Points by Assignee Table
- Custom Charts' Average Age of Issues by Issue Type Bar Chart
- Custom Charts' Cumulative Totals Line Graph
- Custom Charts' Time Spent by Issue Type Funnel Chart
- Standard + Custom Fields

## Recipe Six

### Dashboards that don't just count issues

#### Method

Almost all of the native Jira dashboard gadgets count issues: how many issues are in a specific status, how many are assigned to a specific user, etc. But does issue count tell you everything you need to know to make decisions with your Jira data? In a lot of cases, no. There are several other very valuable metrics you should be reporting on if you want to tell your teams a good, solid story about what's going on.

First, add [Custom Charts for Jira](#) or [Custom Jira Charts for Confluence](#) to your Jira dashboard or Confluence page. If you're in Jira, insert the Custom Charts gadget into your dashboard. If you're in Confluence, insert the Custom Charts macro into your page. (After you've done this, Custom Charts works in exactly the same way in both Jira and Confluence.)

To configure your charts to count things other than issues, simply select your Source and chart format and open the Calculate dropdown. Use the second dropdown to sort through available fields and date comparison options, and then choose whether you want the sum or average of that data. If you select the date comparisons like Time Since or Time Until, more dropdowns will appear to allow you to pick which date fields you'd like to look at.

The following charts combine to create a compelling and highly visual dashboard that tells a clear story about how work is being broken down and completed.



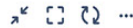


## Custom Charts

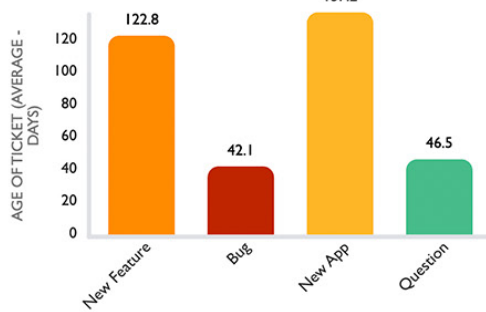


#	Issue Type	Time Spent (Hours)	%
1	Story	144.5	75.3%
2	Bug	32.5	16.9%
3	Task	15	7.8%
Total		192	100%

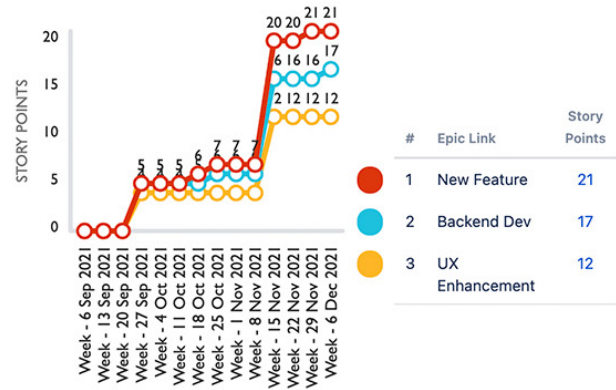
## Custom Charts



Average Age by Issue Type



## Custom Charts



## Custom Charts



Story Points by Assignee

Assignee	Sprint 1	Sprint 2	Sprint 3	Total
Courtney	7	8	14	27
Rebecca	5	-	-	5
Total (Story Points)	12	8	14	32

Continued...



Difficulty level  
Easy



Serves  
Agile Teams

## Recipe Six

### Dashboards that don't just count issues - continued

#### Method

A Story Points by Assignee table shows the number of story points completed in each sprint by each team member, which is much more useful than counting issues because not all issues are sized the same.

An Average Age of Issues by Issue Type bar chart could be used to show the average time since the issues were created, broken up by issue type. This lets you see how long tickets are remaining in your backlog, whether the backlog is getting stale, and how the team is doing with the most urgent ticket types such as bug fixes. (You can also use Custom Charts to calculate time until a specific date, time between two dates, and time to resolution.)

Use a Cumulative Totals line graph to track progress in terms of story points completed per epic and understand how larger bodies of work are progressing. (You could also use a Cumulative Totals line graph to track how much budget is being spent over a period of time.)

Put the icing on the dashboard with a Time Spent by Issue Type funnel chart. This tells teams that track their time in Jira what kinds of issues are taking the most or least time, which is especially helpful when planning future work.



## Filter Results: Blockers

T	Key	Summary	Status	Assignee
	JSDSUP-57	Add support for MasterCard	REOPENED	Unassigned
	JSDSUP-41	Integrate with Google Pay	IN PROGRESS	admin
	JSDSUP-39	VPN access request	IN PROGRESS	admin

1-3 of 3

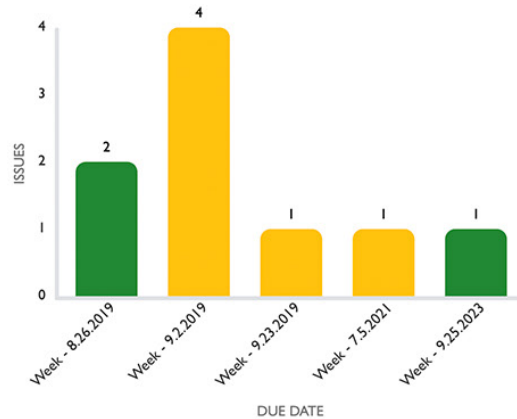
## Custom Charts

## Origins of Issues Added after Sprint Start

#	Sprint	Project A	Project B	Total
1	Sprint 1	15	-	15
2	Sprint 2	15	-	15
3	Sprint 3	14	-	14
4	Sprint 4	4	-	4
5	Sprint 5	-	7	7
Total (Issues)		44	7	51

## Custom Charts

## Issue Completion by Due Date



DUE DATE

Total: 9

Did we hit our due date?

Not completed by due date | 6 Completed by due date | 3



You will need:

- Adaptavist's Scriptrunner
- Knowledge of Jira Query Language (JQL)
- Custom Charts for Jira
- Standard + Custom Fields



Difficulty level  
Medium



Serves  
Jira Admins,  
Power Users

## Recipe Seven

### JQL-infused dashboard for Scrum Masters

#### Method

To make a more targeted dashboard that Scrum Masters will find particularly palatable, start exploring the world of advanced Jira Query Language (JQL). As a Scrum Master, you're likely looking for answers to specific questions like, why weren't we able to complete all of our work on time? Or how can I easily identify blockers and dependencies on other teams? JQL allows you to write more complex queries and find answers by narrowing down the issues.

JQL queries can be saved in Jira as filters. These filters become extremely powerful tools when you're building dashboards or doing any other type of reporting. Most native Jira dashboard gadgets (as well as ones from marketplace apps) rely on saved filters to determine which issues display in each one.

If you've ever tried to build complex JQL queries, you've probably realized that there's a lot you can't do with out-of-the-box Jira. Although powerful, the JQL options that are available by default in Jira are limited. Many people trying to advance their reporting turn to marketplace apps to fill in the gaps. Our favorite option is [Adaptavist's ScriptRunner](#). This comes with some mouthwatering JQL functions that you can use in your saved filters and make a dashboard with just the right combination of flavors.





## Filter Results: Blockers

T	Key	Summary	Status	Assignee
	JSDSUP-57	Add support for MasterCard	REOPENED	Unassigned
	JSDSUP-41	Integrate with Google Pay	IN PROGRESS	admin
	JSDSUP-39	VPN access request	IN PROGRESS	admin

1-3 of 3

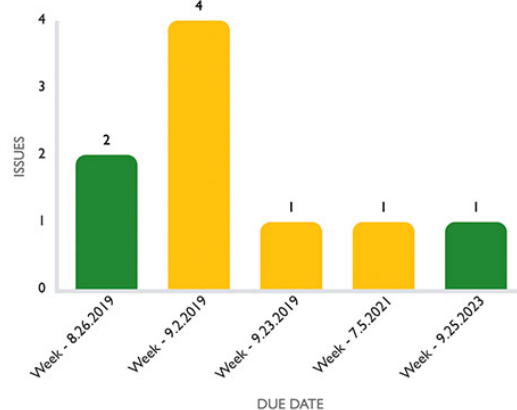
## Custom Charts

## Origins of Issues Added after Sprint Start

#	Sprint	Project A	Project B	Total
1	Sprint 1	15	-	15
2	Sprint 2	15	-	15
3	Sprint 3	14	-	14
4	Sprint 4	4	-	4
5	Sprint 5	-	7	7
Total (Issues)		44	7	51

## Custom Charts

## Issue Completion by Due Date



DUE DATE

Total: 9

Did we hit our due date?

Not completed by due date | 6 Completed by due date | 3

## Recipe Seven

### JQL-infused dashboard for Scrum Masters - continued

#### Method

While there are many options for Scriptrunner functions (and we encourage teams to explore the documentation) there are three we think are particularly tasty. Scriptrunner's `hasLinkType` function adds the ability to write queries based on issue links. This is helpful because the native querying on issue links isn't very strong. Use the `hasLinkType` function for things like reporting on dependencies. Search for issues with a link type of 'blocker' or 'dependency' (depending on what link type you use in your Jira instance), and you'll see all the issues that are blocking work. Save as a filter, and use a native Filter Results or [Custom Charts for Jira](#) gadget to visualize these dependencies for project or sprint management. Dependency reporting can identify bottlenecks and help you analyze missed deadlines by highlighting where delays may have occurred.

Next, build a query using the `addedAfterSprintStart` function, which returns issues added after a sprint's start date and is great for identifying scope creep. It's particularly helpful if you're trying to identify why teams aren't able to complete their committed scope during the sprint. Without this lovely function, there's no way of having scope creep information easily displayed on a dashboard; you would have to navigate to your board and generate a Sprint Report. Better still, use this JQL in a Custom Charts gadget and chart by fields such as 'project' or 'reporter'. Then you'll be able to identify where the issues are coming from.





## Filter Results: Blockers

T	Key	Summary	Status	Assignee
	JSDSUP-57	Add support for MasterCard	REOPENED	Unassigned
	JSDSUP-41	Integrate with Google Pay	IN PROGRESS	admin
	JSDSUP-39	VPN access request	IN PROGRESS	admin

1-3 of 3

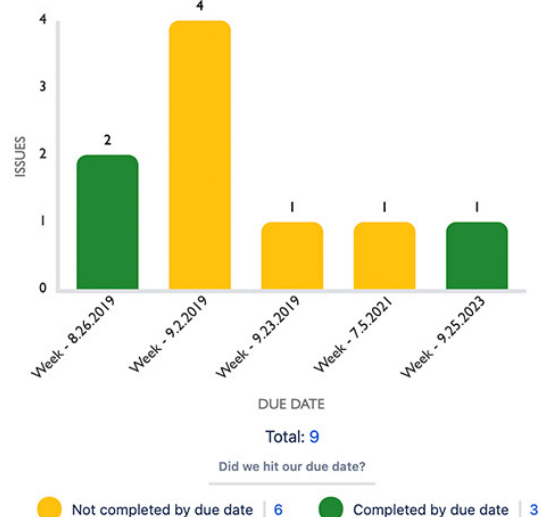
## Custom Charts

## Origins of Issues Added after Sprint Start

#	Sprint	Project A	Project B	Total
1	Sprint 1	15	-	15
2	Sprint 2	15	-	15
3	Sprint 3	14	-	14
4	Sprint 4	4	-	4
5	Sprint 5	-	7	7
Total (Issues)		44	7	51

## Custom Charts

## Issue Completion by Due Date



## Recipe Seven

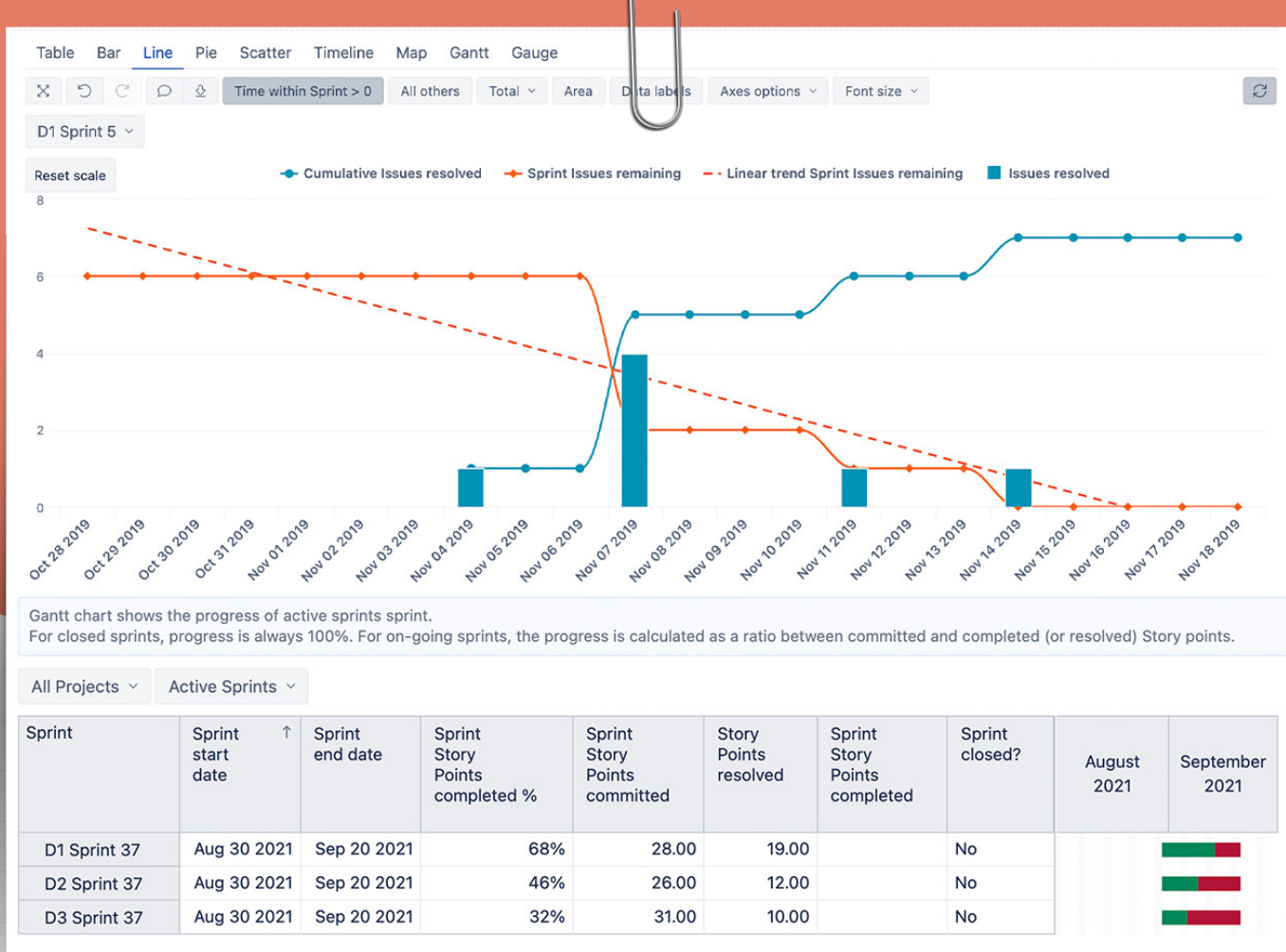
### JQL-infused dashboard for Scrum Masters - continued

#### Method

Finally, use the `dateCompare` function to compare two dates (as long as the dates are on Jira issues). This lets you do things like track release dates, due dates, and planned dates to identify potential issues with your delivery ahead of time. This function could also be used to clean up your Jira data, to see if an issue was updated or resolved before a certain date and identify stale issues. You can also use the `dateCompare` function to filter down to only issues with due dates that are after major milestones. Then in Custom Charts, use the Calculate ➡ Time Between Due Date and Milestone Date to see just how far off they are.







You will need:

- easyBI
- Knowledge of multidimensional expressions query language (MDX)
- Knowledge of multidimensional data cubes
- Weeks of training

## Recipe Eight

### Advanced-level dashboard for project management organizations

#### Method

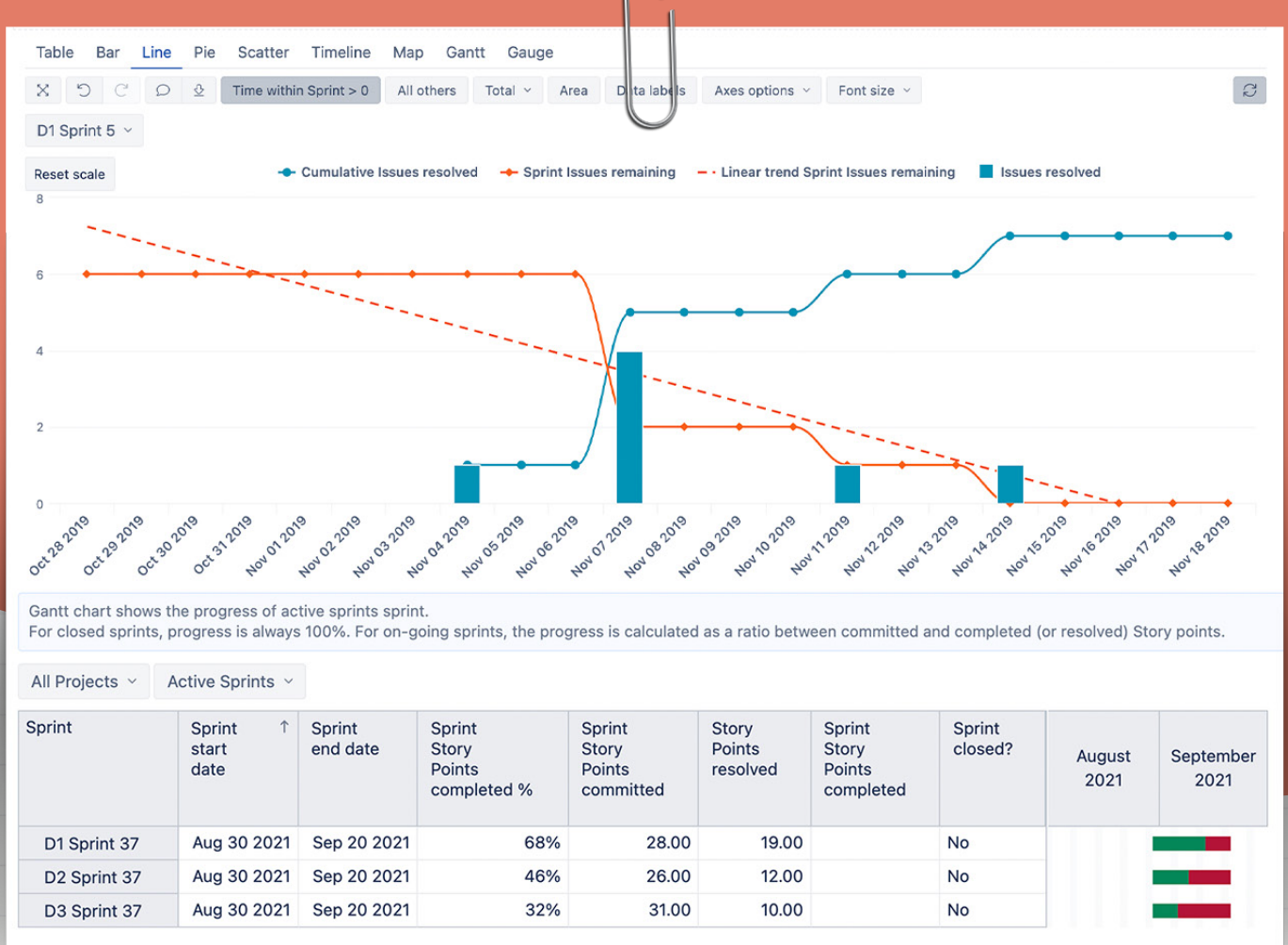
This recipe lets you build a dashboard that's great for project management organizations (PMOs) and project managers working with agile teams. It comprises a Sprint Burndown and a Gantt Chart but, unlike the other dashboards in this cookbook, you have to make these charts in the [easyBI](#) marketplace app before adding them as gadgets to a Jira dashboard.

First, make sure you have a good understanding of multidimensional data cubes and multidimensional expressions query language (MDX). Install and open easyBI, configure your data source to pull in the desired project(s), and begin to select the dimensions for your first report.

For the Sprint Burndown, select the sprint dimension and assign it as a page. Once you've done this, be sure to select the board hierarchy level. Next, drag the time dimension to the rows. As before, define the hierarchy level; we recommend using days. Finally, add the measures dimension to the columns and begin to identify the measures you want in the chart.







Continued...



Difficulty level  
Hard



Serves  
Agile Product  
Teams

## Recipe Eight

### Advanced-level dashboard for project management organizations - continued

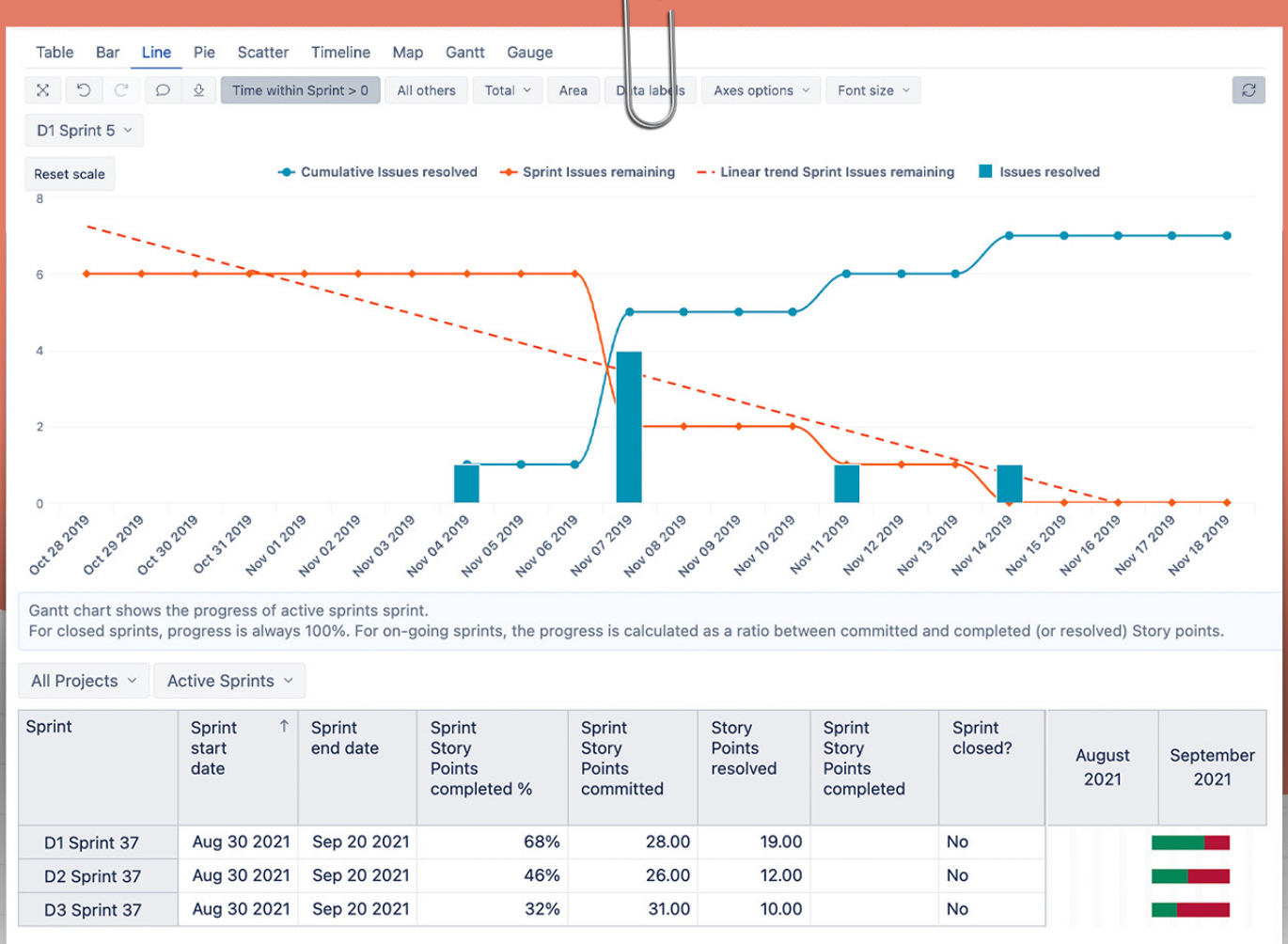
#### Method

This chart requires you to use the predefined Issues Resolved measure, and create two custom measures using MDX. First, define the Cumulative Issues Resolved measure which takes the Issues Resolved and allows you to display the cumulative values over time (in addition to the daily total). Next, create a Sprint Issues Remaining measure that compares the Cumulative Issues Resolved calculation with the issue's status history and sprint dates. From there, you can customize the chart further to display linear trends, sprint filters, and more, to clearly display team progress and projected completion while managing projects.

Next, build the Sprint-Based Gantt Chart. The foundation of this report is the sprint custom dimension which you create and assign to rows, before assigning measures to columns. Once you have added those, add the project and sprint dimensions as pages. For most of this chart configuration, you'll focus primarily on creating custom measures with MDX to calculate the relevant information.







## Recipe Eight

### Advanced-level dashboard for project management organizations - continued

#### Method

Helpful measures to include are Active Sprints, which allows you to filter on sprints that are open. You could also consider Sprint Start Date, Sprint End Date, Sprint Story Points Committed, Sprint Story Points Completed, Sprint Story Points Completed %, and Sprint Completed. Between these eight measures, you'll be able to see a visual representation of your completed and planned sprint work. You'll also be able to filter down the chart to your specific customizations to see forecasted dates based on sprint progress.

Once you've created the reports above, you'll need to add multiple easyBI gadgets to your Jira dashboard in order to display them there.







## Who's hungry?

The Jira dashboard is like a modern and super-well-equipped kitchen, with cupboards full of ingredients for making any number of compelling combinations. Some of these combinations can be prepared ready for consumption in minutes. Others are much more complicated to put together and not for the faint of stomach. Which you choose depends on how often you need to make a ten-tiered wedding cake illuminated with fairy lights and encrusted with hundreds of individual sugar flowers, or if a basic sponge will do just fine.

There are countless more options for slicing and dicing your data on a dashboard; the recipes covered here are only a few.

When whipping up your own recipes, though, be careful not to throw too much in. We recommend a maximum of 6 gadgets per dashboard for ease of use and clarity. Only add more if you're 'taste-testing', i.e. you don't know what to focus on and you need to take a scattershot approach in order to locate a bottleneck somewhere. Generally, it's much better to make different cakes with different flavors, rather than making one cake with every flavor and ending up with something tasting like nothing in particular. Likewise, it's much better to have multiple smaller dashboards with a specific use case for each, rather than one giant dashboard trying to show everything.

How do you choose your ingredients? First define who the audience is and what information they should see. It might be that you want to show the progress of a particular team in a particular sprint, or maybe a multi-project overview of the whole company.

Once you've decided what gadgets you want on your dashboards, you can use them in your status meetings and retros. Individual users can also create personal dashboards with the work assigned to them so they can keep track.

Remember that it's not always possible to get a dashboard right first time. It may not be clear to your audience why the data you've included is relevant, in which case, take their feedback and change the recipe. It might take a few goes to get it tasting just right, but hey, that's what agility's all about.

Visit [oldstreetsolutions.com](https://oldstreetsolutions.com) for more recipes, and *get cooking!*



## About the authors

These recipes were brought to you from the Old Street kitchen by our two chefs, Christopher Berry (Head of Content) and Morgan Folsom (Product Manager). When they're not cooking up dashboards, Christopher's writing time travel novels from his humble abode in the Hampshire countryside, and Morgan's taking strolls in the Seattle rain with her pup. They both enjoy cooking stuff you can, you know, eat too. Christopher's signature dish is Swedish meatballs while Morgan makes a mean risotto.

## About Old Street Solutions

Old Street Solutions is a collection of enthusiastic chefs who bake Jira and Confluence add-ons designed to make the Atlassian kitchen an easier place to maneuver around. Their software helps empower everyone to be competent Jira and Confluence bakers without having to become masterchefs. Their apps include External Share for sharing content beyond your instance; Planning Boards for organizing your work in Jira with more freedom; and Custom Charts for telling powerful and highly visual data stories. Custom Charts was the fastest-growing paid app in the Atlassian Marketplace in 2020.



Platinum  
Marketplace Partner



Custom  
Charts



External  
Share



Contract  
Signatures



[oldstreetsolutions.com](https://oldstreetsolutions.com)

128 City Road, London, EC1V 2NX