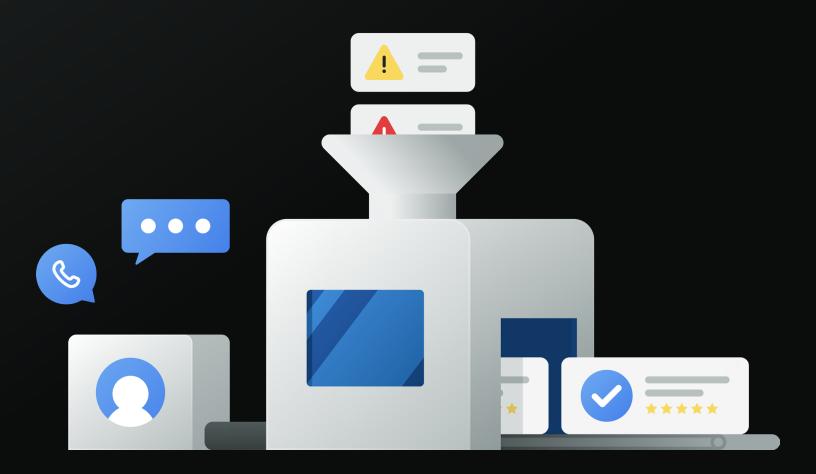




The ultimate guide to powering up your service desk

Expert advice (and 10 bonus tips!) for running an amazing service desk with Jira Service Management.



Contents

Intr	oduction 4	ŀ
Wha	at does a service desk do?	5
Serv	vice desk vs help desk: What's the difference?	ō
Wha	at are the core functions of a service desk?	6 's the difference? 6 a service desk? 7 and help desk tools 8 int? 8 int? 8 int? 7 in 10 in 11 in 13 in 17 in 18 in 18 in 19
The	importance of service desk and help desk tools	
Why	use Jira Service Management?	3
	What is a customer?)
	Jira vs. JSM)
	Jira issues and JSM requests	ı
	JSM top features	3
Mar	naging help through service requests	7
	Fulfilling a service request	7
	Service requests in JSM	3
	Service request workflows)
Get	the word out with canned responses24	ļ
	Canned Responses Pro Templates for Jira	1
	Use Case	5
	Requirements	5
	Set Up	5
	Result)
	Bonus! Sample wording)

Modernize your workflow and automation	
Approval workflows	
Conditional approvals	
Service request automation	
JMWE	
Approval workflow best practices	47
SLAs and continuous improvement	49
SLA Reports	52
SLA Automation	53
Troubleshooting	56
JSM reporting	59
Custom reports	63
Queues	63
Customer feedback	65
Do your research	67
Feedback requirements	67
Tools for collecting customer feedback	69
Customer feedback best practices	78
JSM project templates	
Cloud Templates	79
Server and Data Center templates	81
Communication strategies for JSM implementation	82
Example scenario	83
Translate requirements to JSM instructions	
Checklist - "Rachel's 10 Tips for Managing Support in Jira Se	ervice
Management	00
Resources	93

Introduction

Implementing your service desk is the kind of thing you really want to get right. But even if you hit a few snags, you can get back on track. You just need the right approach, the right tools, and a few tips and tricks. And you're in luck! This handy guide has everything you need to set up your service desk so you can start turning tickets around in no time. (Well, *almost* no time!)

The best way to manage support requests from internal or external customers is through a robust process, a system to track requests, and a knowledge base to educate users. Adopting standard processes sounds simple enough, but it takes a lot of planning to make things run smoothly.

Don't worry, we got you! This guide includes tools and best practices to help you on your service management journey.

I'm Jira expert Rachel Wright, certified Jira administrator and author of the Jira Strategy Admin Workbook and the The Ultimate Guide to Jira Migrations: How to Migrate From Jira Server to Data Center or Cloud. I've teamed up with my friends at Appfire to help you implement sound support processes so your organization can deliver clear and measurable business value.

In this guide, we'll share service industry best practices. These tips and tricks will help you turbocharge your service desk and use Jira Service Management (JSM) to collect, fulfill, and track support requests, incidents, problems, and change requests.

We'll also show you how to extend Jira with apps from the <u>Atlassian Marketplace</u> to automate manual steps, create templated support responses, collect customer feedback, and more!

Whether you're new to Information Technology Service Management (ITSM) or simply want to level up your current processes, this guide can help.

Oh, and one more thing: ITSM isn't just for technology teams! Non-IT teams like HR, sales, facilities, and legal can use these same ideas to deliver excellent service to *their* internal and external customers, too.

Now let's dive in!

Who's this guide for?

- Organizations adopting Information Technology Service
 Management processes or transitioning to Jira Service
 Management from another service or support tool, as well as
 businesses undergoing a digital transformation, like when moving
 to the cloud
- Application administrators who configure and maintain Atlassian applications like Jira Service Management
- Application owners, leaders, or executives who direct IT strategy, service, and support procedures
- Change advisory board members or change managers who assess, approve, and schedule change requests
- Help desk support agents who receive and resolve support requests in Jira Service Management
- Help desk support managers who monitor progress, SLA adherence, and customer satisfaction
- Developers who collaborate with support staff to answer questions and solve problems
- Knowledge managers or technical writers who craft selfservice help content

What does a service desk do?

A service desk is the single point of contact between a service provider and their customers. A typical service desk will manage incidents and service requests, while also handling communication with users. In essence, a service desk is a communication center where customers (both internal and external) can find help from their IT service providers. Because everyone needs a little help sometimes!

Help comes in many forms, and can include incident resolution or service request fulfillment. Regardless of the type of help being provided, a service desk aims to deliver high-quality service to customers quickly and efficiently.

Service desks manage multiple activities, including ITSM actions like service request management, incident management, knowledge management, self-service, and reporting. In an IT service desk, teams can assist customers with incident resolution, as well as service request management. All in all, service desks do more than answer questions: they are robust, service-focused, customer centers delivering solutions to all customers.

Service desk vs help desk: What's the difference?

You might be thinking "we're running a help desk, isn't that the same thing as a service desk?" While there are some similar functionalities, there are also significant differences between the two.

A help desk is meant to provide technical support to end users, troubleshoot customer and user issues, and guide end users through specific tasks or actions. A knight in shining armor of sorts!

A service desk, on the other hand, looks at the business's needs as a whole, instead of focusing solely on addressing user needs. By taking the broader business perspective into account, a service desk works to improve the efficiency and performance of ITSM at the organizational level, meeting service level agreements (SLAs) and shaping the ways IT professionals provide service both to internal employees and external customers.

Within most organizations, the help desk is merely a component of the service desk. This guide will help you power up all aspects of the service desk to better reflect your business's mission and values.

What are the core functions of a service desk?

Everyone loves a list. So, let's break down some of the core responsibilities of a service desk:

- Managing requests, problems, and incidents
- Addressing any IT concerns within all departments of the organization
- Tracking customer issues
- · Tracking adherence to response and completion goals
- Identifying key characteristics of reported incidents and routing them to the appropriate stakeholders.
- Monitoring reports and metrics including:
 - First-call resolution rate (FCR)
 - ° Repeated tickets
 - ° Incident volume
 - ° Ticket trending
 - ° Ticket backlog

A well-rounded set of processes can really amp up your organization's efficiency. With the right integrations of IT management tools, your service desk can use automation to help provide seamless execution of IT operations within the organization.

The importance of service desk and help desk tools

When getting your service desk up and running, choosing the right software is critical, because that tool becomes the foundation of your ITSM strategy. Your service desk serves as the interface between the customers and the IT team. As the backbone of your organization, support teams deserve software tailored to their unique needs.

Features like knowledge management and reports are crucial to service success, but simple setup and intuitive apps make it easier to communicate with customers, connect to critical monitoring tools, measure customer sentiment, and have continuous quick access to metrics. Don't underestimate the value of easy to use software! With a dedicated service desk management tool, your support team can be more transparent, collaborative, and efficient.

Jira Service Management ("JSM") is uniquely positioned to provide those capabilities and more. JSM also facilitates communication between other teams using Jira, like development and operations. A real win/win for everyone involved!

Why use Jira Service Management?

Top-performing teams need top-performing software to manage the neverending requests for service and support. In 2013, Atlassian, the maker of other project management and collaboration software, launched Jira Service Management, or JSM for short. It was originally named Jira Service Desk.

Before JSM launched, <u>Atlassian reported that 40% of customers were</u> adapting <u>Jira to handle support and related requests</u>. I was one of those customers. I created a dedicated <u>Jira project</u> for support work and a custom dashboard gadget to direct users to the right place for their requests. It was a "<u>MacGyvered</u>" version of today's JSM customer portal.

JSM is not just a "ticketing system." Atlassian's goal for this application is to "bring IT operations and development teams together to collaborate at high velocity and power digital enterprises." JSM has grown up a lot since its launch, and I'm excited to share the many ways it supports ITSM.

IT Service Management (ITSM) refers to the processes that allow businesses to design, build, deliver and support IT services. As businesses evolve to become more technology dependent, ITSM continues to be crucial to supporting teams and enabling them to deliver value.

Using JSM can help your service team to better handle the complex responsibilities of modern ITSM in an increasingly digital environment; cross-team collaboration, prioritizing and fulfilling requests, managing incidents and system outages, tracking assets, and so much more. And because JSM is an Atlassian product, it seamlessly integrates with related products and services so everything can be managed from one place. JSM's a real superhero tool of the ITSM space!

What is a customer?

Regardless of department, we all have "customers." Whether it's an external client, an internal team member, or users of our organization's products and services, we all have someone to support. Some of us support multiple types of customers without even realizing it!

Picture a Human Resources generalist at your company. Who is that person's customer? The obvious answer is you and other employees. But there are also other types of customers counting on HR's help: job applicants, hiring managers in other departments, and even fellow HR team members.

What types of customers need help from you or your team? **Make a list** of all the potential users or groups, the type of help they need, and the expected request frequency.

Does your sales or marketing team use personas to understand customer wants and needs? You can use a similar approach when crafting or enhancing IT support processes. Once you've defined your customer, you can design your process (and your JSM project).

Now that you understand your audience, how will you manage their wants and needs? The answer: through your service desk, powered by Jira Service Management.

Jira vs. JSM

Jira Service Management is built on Jira's foundation and integrates with Confluence as a knowledge base. It's an obvious support solution for organizations already using other Atlassian software.

In the past, teams like IT, development, and operations all used separate, disconnected software. There was no easy way to share or sync information. Details were often lost or duplicated, and there was no uniform information tracking strategy. But now we have Jira and JSM: no need for multiple vendors and tools.

Jira provides powerful issue tracking, workflow, and reporting functionality. JSM adds additional service-specific features right on top of Jira's foundation. JSM extends Jira so your organization can effectively manage incidents, problems, changes, service requests, and more.

Application Differences

Jira Features	Jira Service Management Features
IssuesWorkflowReportingAutomationAnd more	Jira features plus: Request forms Queues SLAs Self-Service portal Knowledge base integration And more



Aappfire

What's the difference between Jira Service Management ("JSM") and Jira Software?





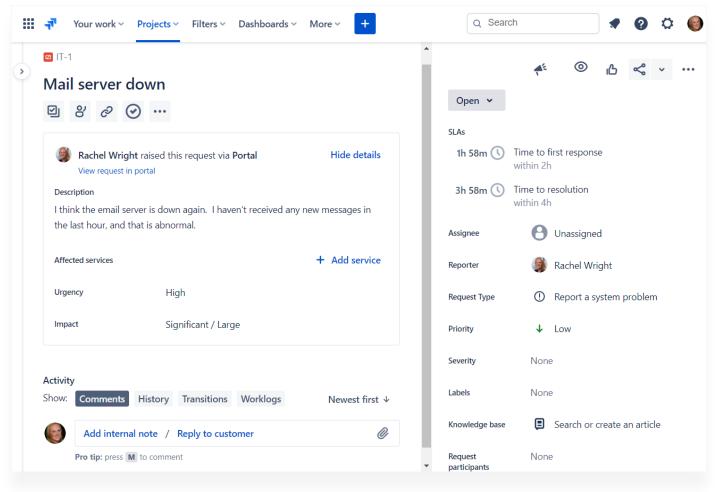
Preseted by Kerry O'Shea Gorgone with guest Rachel Wright

Watch this 8 minute chat about the different features each tool has that makes it suited for different types of work.

Jira issues and JSM requests

Jira and JSM have many overlapping features, but it's important to distinguish between JSM-specific actions and Jira terms. An "issue" is an individual item in Jira. It's identified by a unique key in the format: ISSUE-1. A Jira issue represents many things, like a bug report, a task, or a user story. A "request," however, is a simplified view of issue data in the Jira Service Management customer portal.

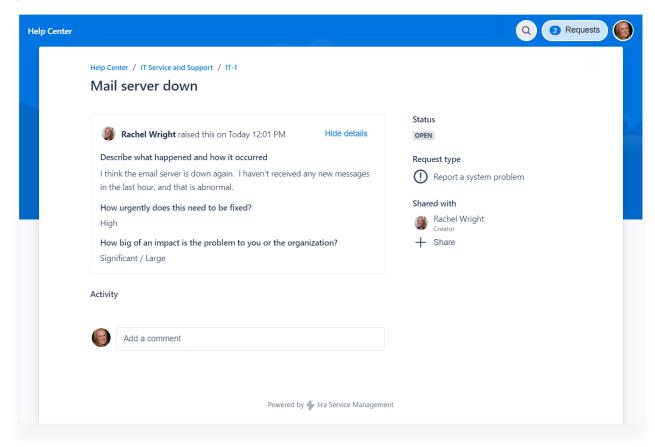
Here's an example of a trouble report issue for a problem with the mail server.



Issue IT-1 in Jira Cloud

This screenshot shows how the issue looks in Jira.

The next screenshot shows how the same information looks as a request in JSM.



Request IT-1 in the Jira Service Management Cloud customer portal

It's important to know that both views represent the exact same trouble report. The database contains only one unique record for it.

Both screenshots contain the same information, but the display is different depending on the audience. The comprehensive Jira issue view is for support team members, while the simplified JSM request view is for customers.

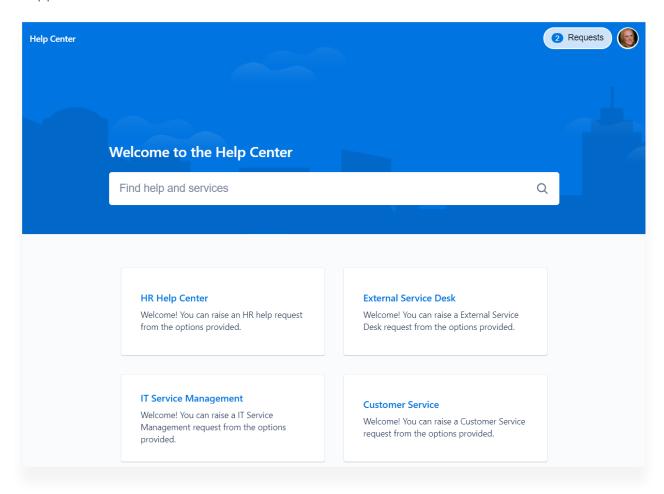
JSM top features

JSM extends Jira's capabilities by adding support-specific features for customers, support agents, and support leadership. Here are some of the top JSM features:

Self-service portal

The JSM self-service portal (sometimes called the customer portal, help center, or service desk) is where employees and customers submit requests. It's a simplified interface with easy to complete web forms. This is my favorite JSM feature because it makes the request entry process incredibly simple for the customer or end user. Customers don't need technical skills, Jira knowledge, or licenses to enter a request.

Organizations can customize selected portal branding features, like colors, logos, and copy. You can even have multiple service desks or areas of support, as seen in the screenshot.



Multiple service desks in the same customer portal

Knowledge base

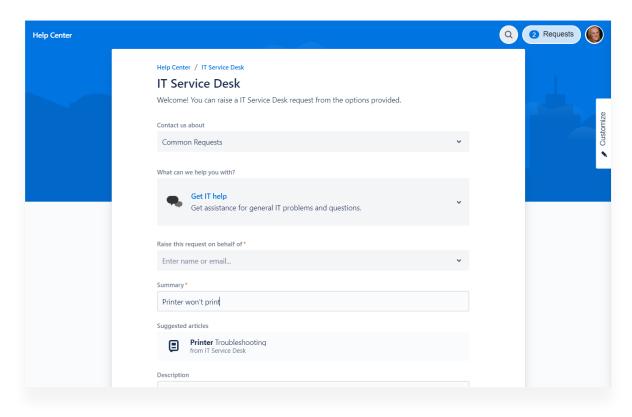
Do your customers frequently ask for the same types of support? Wouldn't it be nice if customers could solve a request by themselves, *without* assistance from the support team?

That's exactly what the knowledge base integration is for.

Just connect a Confluence application and create your own self-service help center where customers (and agents) can access frequently requested information, "how to" articles, step-by-step instructions, and much more.

Here's how it works: The customer searches for help or starts creating a request. If the information entered contains keywords or phrases also present in help content, those are automatically shown.

In the example shown here, the customer entered "Printer won't print" in the summary field. A Confluence article titled "Printer Troubleshooting" automatically appears. If the article solves the problem, there's no need to continue creating the support request. The customer gets a fast resolution without unnecessary human interaction. That's a win for everyone.



Suggested help articles provided by Confluence

ITSM support types

JSM is specifically designed for common ITSM support areas and activities. They are:

- Service management
- Change management
- Incident management
- Problem management
- · Asset management
- Configuration management
- Knowledge management

Service level agreements

A service level agreement ("SLA" for short) represents a goal or a commitment between a service provider and a customer. For example, a request to replace a broken monitor might have a fulfillment goal of one business day. A request for an upgraded monitor however, might have a completion goal of one week. The goals in this example are based on urgency. A broken screen prevents an employee from working and should be addressed quickly. A desire to upgrade a monitor is less urgent.

SLAs allow teams to define goals, prioritize requests, and track metrics. Common metrics include time to first response, time to resolution, time waiting for customer input, time waiting for vendor action, specific goals for VIP customers, and other success factors.

Don't worry. More on SLAs to come in this guide!

Additional JSM features

- Project templates with custom issue types, sample request types (forms), and example workflows
- Custom request types for additional support activities and areas not already included
- Customizable language so field names and workflow statuses in the portal are customer friendly
- Queues to categorize related requests. A JSM queue is like a Jira filter except it's shared with the entire support team.
- Automation to remove repeatable manual steps like resolving inactive requests, reminding customers to provide more information, or automatically assigning requests based on location, skill set, or availability
- Additional reports for measuring workload, customer satisfaction, and other support-specific insights
- Additional communication features like customizable notifications, requests via email, an embeddable form for your website, and chat support
- A form builder to collect additional or conditional information
- A huge <u>marketplace</u> of apps, connections, and integrations to further extend capabilities
- And more

Managing help through service requests

We all need a little help sometimes. For customers and employees alike, being able to submit an inquiry to a professional for assistance is essential to getting things done.

In a service desk setting, people ask for help in the form of a "service request," a record of a need or desire from any type of customer. It's commonly used for something new a customer needs but doesn't have.

For example, new hires generally require a standard list of devices, software, and access. Service requests can gather and track these items to make sure new employees have everything they need for their first day of work. This request type frequently supports standard, common, and recurring needs, so these requests are usually low risk, and may even be fulfilled using automation.

Just a reminder! A service request's specific purpose is to ask you for *help*. Don't confuse service requests with incidents, changes, or problems.

Although service requests are sometimes related to other request types, each has its own distinct elements, processes, and issue types. Separating service requests from other requests is one way to help the support team categorize, triage, and manage work.

Fulfilling a service request

To set your team up for success, you'll want to establish a service request process. The first step is understanding how filling a service request works.

A service request is generally received by a support, service, or fulfillment team. When requests are received, a support team leader may review them, then assign them to the proper resource based on skills, knowledge, availability, location, and many other factors.

Sometimes help from other teams is required to complete a request. Jira Service Management calls these collaborators "request participants." Request participants don't have as many permissions as agents, but they can view, comment on, and receive request notifications.

In order to successfully fulfill requests in a timely manner (and be able to scale over time), you need software like JSM.

Service requests in JSM

Jira Service Management helps teams distinguish between requests for help or "service" and other types of requests. Different request types often have specific custom fields, workflows, and settings.

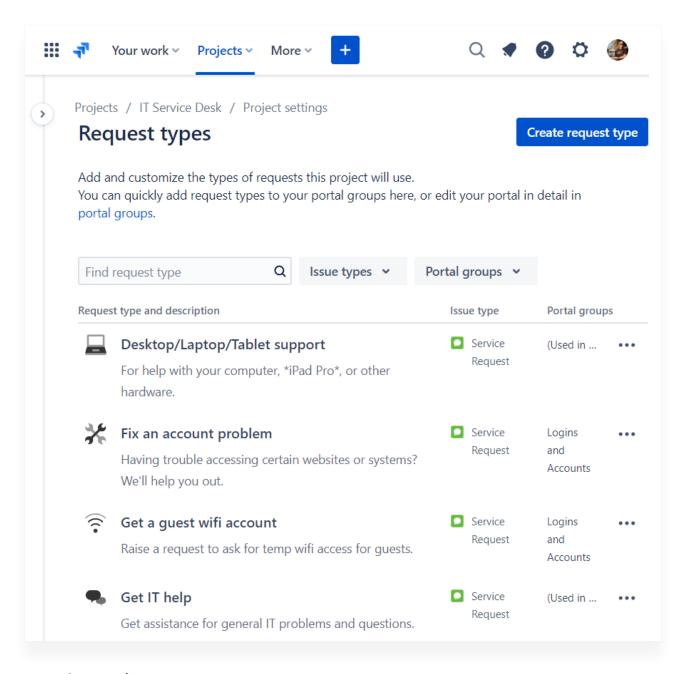
For example:

Scenario	Jira Configuration Area
A <i>service request</i> for a new company phone may require an additional approval process. An <i>incident</i> may not.	Workflows
An <i>incident</i> may collect and store root cause information. A <i>change</i> request may not.	Custom fields
A certain type of <i>problem</i> may automatically be assigned to a specific team member. An <i>incident</i> may require an initial investigation before assignment.	Components or Apps

By segmenting requests based on type, the request uses the most efficient process and collects the right type of information, without sacrificing reporting capabilities.

For instance, how would you calculate the efficiency of a service request that took 30 minutes to complete vs. a change request that took 30 days to complete? If all requests are managed the same way, it's hard to compare and evaluate the effort for these very different tasks. But segmenting requests enables you to compare "apples to apples" so you can draw meaningful conclusions about how to refine your processes.

Segmenting also helps leadership understand recurring requests, compare similar data points, and allocate the proper amount of resources and budget based on request volume.



Example IT service requests

In short, JSM is going to manage your service requests in a way that will ensure your service desk can see all the important details.

Examples of service requests include:

- Hardware or software support
- Employee onboarding
- General help or questions
- · Anything a customer needs but does not have

Common requests for help include:

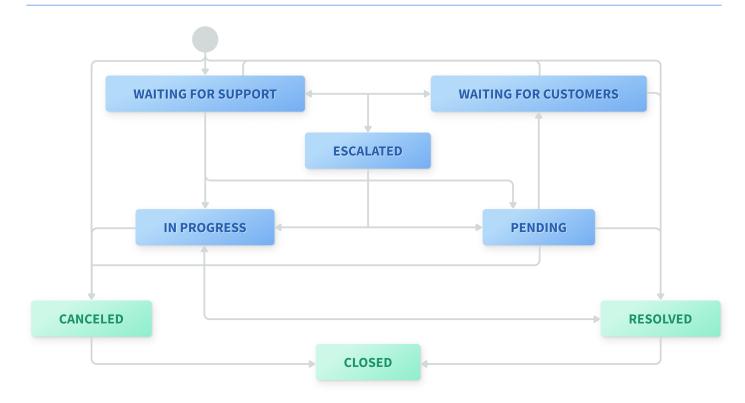
- An action (like a password reset)
- Advice (like how to handle a suspected computer virus)
- Information (like clarification on the reimbursement policy)
- Support (like assistance retrieving a blocked email message)
- An asset (like a new software license or a desk chair)

Service request workflows

You understand the importance of JSM for handling requests, but there are other aspects of service requests that need to be addressed in order to have the full capabilities of a powerful service desk, including service request workflows.

Service project workflows generally contain steps that represent the backand-forth nature of support work. Service work isn't always linear and the workflow doesn't always follow a prescribed path.

Completing a request might involve one or more conversations with a customer, management approval, or action by a third-party vendor. Requests may also need to be escalated or deescalated. That action could occur in multiple places in the process, like when a request is awaiting support or awaiting customer input.



Default service request workflow

Here's one of the default JSM service workflows. This example is commonly used for service requests and includes statuses like: waiting for support, waiting for customer, pending, resolved, in progress, canceled, escalated, and closed.

As shown in the diagram, the path a service request takes during its life cycle varies depending on an infinite number of circumstances. Multiple incoming and outgoing transitions are common in support request workflows. For example, requests in the "In Progress" status can move to multiple other statuses as needed.

How do service requests differ from other requests?

If you're new to ITSM or to providing support in Jira, you may be asking yourself what the difference is between service requests and other types of ITSM requests. Below, we'll quickly break down the hierarchy of requests and how service requests differ from others you might come across in your service desk.

The default JSM issue types are:

Incident. An incident is generally defined as an unplanned interruption to an IT service, or something that reduces the quality of an IT service. When an incident is created in a service desk, it's most often categorized as a break/fix issue.

Problem. While problems are related to incidents, they're different because they're characterized as something that is the *cause* of one or more incidents. The cause may not be known at the time the problem is recorded, but often needs further investigation to resolve.

Change request. A change request is a proposal to alter some aspect of a product or system, either internally or externally. For example, a server upgrade, a configuration change, or a modification to code in production.

Service requests. Incidents and problems deal with needs. A service request deals with *wants*. Some service request tickets are not as urgent as incidents and problems. These can be scheduled and planned around more urgent requests.

Service request best practices

- · Make the request process as easy as possible for customers and agents
- Create multiple ways to accept requests, communicate with customers, and collect feedback
- Determine how your team or organization defines success (and how you'll measure it)
- Keep the requestor happy and engaged by providing regular updates (even if there's no update)
- Don't ask for information or feedback that won't be used

What other ways can your organization improve the service request process? Which organizations are known for great customer support? How can you incorporate some of their process into yours?

Q Bonus ideas

Use the Confluence knowledge base integration to suggest selfservice and how-to information

Try the built-in JSM "email requests" feature (It's better than Jira's "create via email" function.)

If you have external customers, customize the look of the customer portal and the voice used in email communications

Get the word out with canned responses

Improving resolution time pays off for your team, your company, and your customers. Giving agents access to premade, templated responses to tickets gets your team from ticket to resolution faster.

With automation, you can set up templated responses that help service desk representatives answer common questions more quickly. Elsewhere in this guide [LINK], we'll cover ways to automate actions in response to common requests without human intervention. But in many cases, a human is best qualified to understand customer requests.

Even at large organizations, some support teams are still manually copying and pasting answers to their customers. If you tally up the number of hours spent copy/pasting, the total is substantial. And that's if you have predefined answers available somewhere. If you haven't taken initial steps yet, now's a good time to start.

Having predefined answers to common requests makes it much easier for agents to answer questions consistently and quickly, streamlining your processes. And customers get the same level of service, no matter who handles their request.

Minimizing the tedious copy/pasting also makes your agents happier, which improves productivity and retention. With automation, they spend less time on repetitive tasks, freeing them up to focus on more creative, satisfying work. That's something everyone will appreciate!

Canned Responses Pro Templates for Jira

The <u>Canned Responses Pro Templates for Jira</u> app makes responding to customer requests quick and easy. Create canned responses for common service requests with reusable and dynamic templates, as well as message signatures for added consistency and professionalism. Automation sets your agents up for success and you can even automate a series of actions for common issues in Jira and Jira Service Management with Canned Responses.

With Canned Responses, your agents can create a delayed message, or schedule a comment to make sure things don't fall through the cracks. For example, say an agent sends a fix to a client and needs to check back in three days to confirm that the solution worked. Rather than setting a calendar reminder or tickler system, the agent can schedule a delayed message to go out in three days asking "Did the fix work for you?"

With just a couple of clicks, your team can save a significant amount of time, achieve higher productivity, and improve time to first response ("TFR").

Use Case

The sales team uses a mobile app to access customer information when traveling to client locations. But the mobile app frequently malfunctions. Luckily, the instructions to remedy the problem are usually the same. The support agent only needs to choose whether to provide instructions for an Android device or an iOS device. (Hopefully the device type is collected in the initial request!)

Let's create some reusable instructions for agents to use when servicing mobile app problems.

Requirements

You'll need the following:

- Access: Application administrator permissions (to <u>install the Canned</u> <u>Responses app</u>) and access to related Jira projects and issues
- Deployment type(s): Cloud, Server, or Data Center
- Procedure: Install the app from the "Find new apps" admin page in your Jira application or from the Atlassian Marketplace. In Cloud, a trial subscription is automatically started. In Server or Data Center, apply a free trial or paid license on the "Manage apps" admin page.
- Prerequisite(s): Reusable wording for testing support scenarios. See some pre-written examples below.

We'll try out the following app features:

- Create new reusable wording templates
- Use variables in the response copy
- Automatically transition issues when a template is used
- Create a personal signature template

Set Up

One of the first problems I always encounter is how to access the settings for the app I just installed. There's usually a button, a menu, or a page to locate. Start by looking in the following areas:

	Typical Locations	Used by this App?
1	The left sidebar in the "Apps" admin area (you may need to refresh the page)	✓ Yes
2	The app's listing on the "Manage apps" admin page	⊗ No
3	The "Apps" menu in the top navigation bar (Cloud)	⊗ No
4	The left sidebar in the "Project settings" area of a Jira project (project administrator view)	⊗ No
5	The left sidebar of a Jira project (end user view)	✓ Yes
6	The toolbar under the "Summary" field of a Jira issue	✓ Yes
7	A panel at the bottom or right side of a Jira issue	✓ Yes

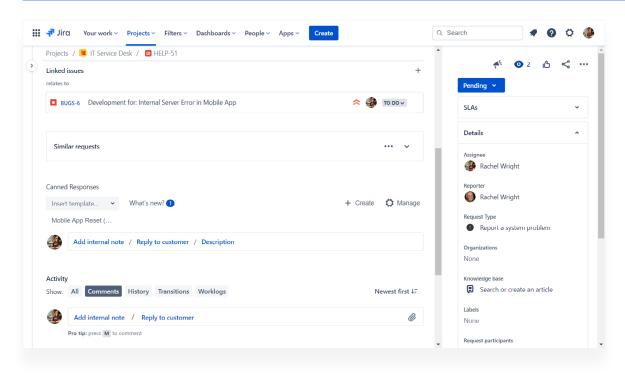
Still can't find the settings or where to start? Check the app's documentation for guidance.

Implementation

After installation, visit any settings areas to see what options are available and make any desired changes. I typically leave the default settings "as is" at this early testing state.

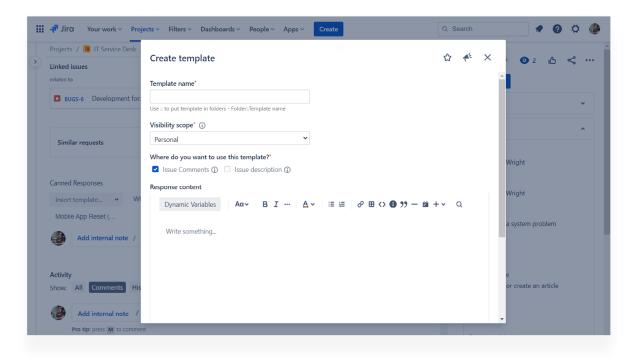
The app is immediately available to use and includes sample templates to get started. Here's how I created my first custom response template.

• Find the "Canned Responses" panel at the bottom of any Jira issue



The "Canned Responses" create button

• Click the "Create" button on the right to create a new reusable template



The "Canned Responses" template creation overlay

- In the overlay, enter a unique and descriptive template name
 - ° Paste in the desired message content

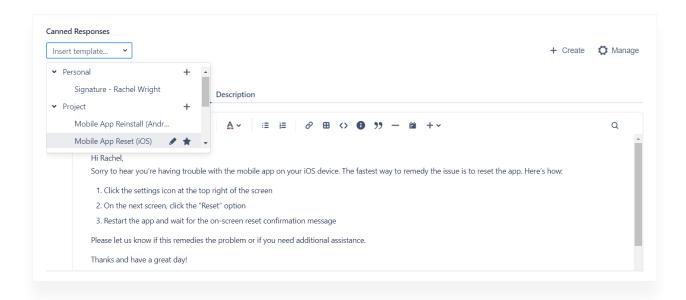
- Use the options in the toolbar to insert variables, format the copy, and more
- ° Then click the form submission button

For my exploration, I created two sample templates and limited their scope to a particular Jira ITSM project. I also created another template for my personal signature.

Finally, go to any Jira issue and choose a template to instantly insert the canned message.

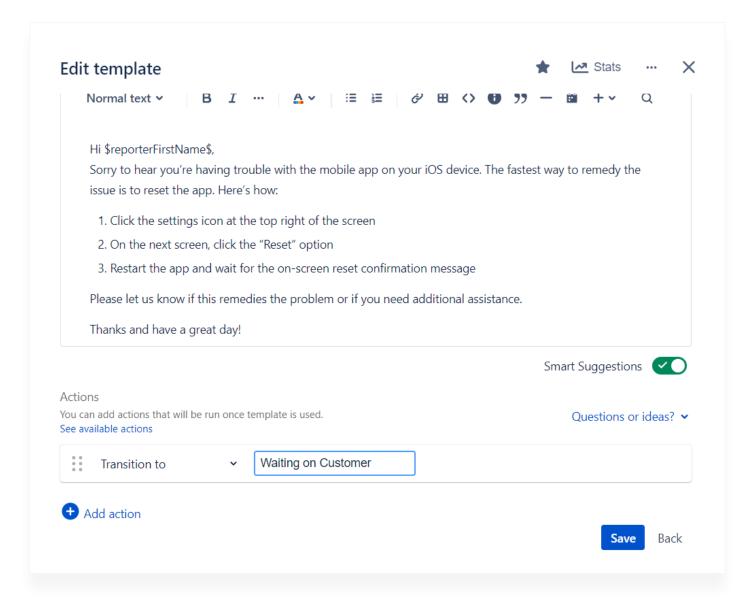
Expert tip!

It's possible to insert content from more than one template. For example, after inserting the desired canned message, you can also insert the agent's personal signature, if desired. This structure is super flexible and there are lots of communication strategies and template combinations to consider.



One personal template and two project templates

There's another useful template function I should mention. Look for the "Actions" area at the bottom of the template settings screen. In the screenshot, I've added a "transition to" action and specified the "Waiting on Customer" workflow status. Any time this template is used, the issue will automatically transition to the specified status with no manual effort needed. Nice!



Additional actions in the "Canned Responses" app



This app isn't just for Jira Service Management! Use it to create reusable wording for other Jira applications and project types too.

Result

Before this app, I stored reusable wording templates in Confluence. Storing the copy in Jira (where I ultimately use it) is so much faster! Once the templates are set up, there's no need to copy, paste, or do a bunch of other manual activities. The entire communication process is handled with one click. Agents are happy because there's less effort required and leadership is happy because standard and consistent messaging is used. Win/win!

Expert tip!

Not a writer? Ask a customer service rep, technical writer, or someone on the marketing team to help create the standard wording. They'll appreciate that the company brand is properly reflected in the responses.

This is just the beginning of what this app can do! I hope you'll explore the many additional capabilities in your test application.

Bonus! Sample wording

Here's the sample copy and settings I used for my examples. Feel free to use them for your testing and/or customize them to fit your needs. Check out <u>Canned Responses content in The Hub</u> for more inspiration.

Template name: Mobile App Reinstall (Android)

Sample message

Hi \$reporterFirstName\$,

Sorry to hear you're having trouble with the mobile app on your Adroid device. The fastest way to remedy the issue is to reinstall the app. Visit the company software page to download and reinstall the software.

Please let us know if this remedies the problem or if you need additional assistance.

Thanks and have a great day!

Additional settings:

- Variable(s): Use the format "\$reporterFirstName\$" to insert dynamic information
- Formatting: Includes a link to related information
- Scope: Single project
- Action: Transition to "Waiting on Customer" status

Template name: Mobile App Reinstall (iOS)

Sample message

Hi \$reporterFirstName\$,

Sorry to hear you're having trouble with the mobile app on your iOS device. The fastest way to remedy the issue is to reset the app. Here's how:

Click the settings icon at the top right of the screen

On the next screen, click the "Reset" option

Restart the app and wait for the on-screen reset confirmation message

Please let us know if this remedies the problem or if you need additional assistance.

Thanks and have a great day!

Additional settings:

- Variable(s): Use the format "\$reporterFirstName\$" to insert dynamic information
- Scope: Single project
- Action: Transition to "Waiting on Customer" status





Sample message

<insert blank line here>

Rachel

Rachel Wright

Email: supportteam@awesomecompany.com

Additional settings:

- Format: Separate this content from other copy using a blank line
- Scope: Personal
- Action: Transition to "Waiting on Customer" status

Q Expert tip!

Check out the additional browser extension for <u>Chrome</u>, <u>Edge</u> or <u>Firefox</u>. It improves app usability by integrating the Canned Responses panel with the standard comment area.

Modernize your workflow and automation

Service desks rely on a lot of manual processes, like copying and pasting answers. Even in large contemporary organizations, more modernization of these processes is necessary. If you were to tally up all of the hours spent on maintaining these manual processes in ITSM, the total might surprise you.

That's why automation is the key to improving service desk workflows and resolution time. Atlassian apps extend workflow functionality and help issues progress smoothly, which is vital to the success of your service management.

In this section, I'll discuss the ways automation and workflow settings can boost service desk performance, and explore some apps your support teams might want to try.

Approval workflows

When your team lead needs visibility and the organization needs to understand what's changing, administrators add approval steps to their request workflows. Approvals are most commonly used for change requests and service requests. In Jira Service Management, approvals are not forced by default. If you need to, you can add restrictions to address a wide variety of use cases.

Common approval requests include:

- Granting or revoking permissions (like giving a power user administration abilities)
- Installing an app from the Atlassian Marketplace
- Physical and digital access requests (like a key to enter the executive floor)
- Changes that impact a large number of users (like a server restart)
- Large equipment purchases (like new monitors for everyone)
- · Corporate travel 50 miles from the home office

Approvers are often:

- Members of the change advisory board (CAB)
- A team or department lead
- A requestor's supervisor

Conditional approvals

Organizations often require approval for some requests, but not others.

Approvals are based on:

- Request type
- Affected service
- Change advisory board (CAB) membership
- User input

IMPLEMENTATION OPTIONS

Say there are two request forms: one for a new mobile device and another for a new mouse. Since a new phone is likely to cost hundreds of dollars, an approval process may be warranted. But the mouse request is much less expensive, so it may require a less complex procedure.

There are two ways to handle this. You can simply use a specific issue type, request type, and workflow for expensive items and a different issue type, request type, and workflow for inexpensive items. A more elegant solution, however, is to use the built-in JSM approval feature, apps, scripts, or automation to implement approval requirements.

AUDITING AND COMPLIANCE USE CASE

For organizations with audit and compliance requirements, it's good to make sure approval is granted before work starts, and that the approval step wasn't skipped. Luckily, this is easy to do with <u>Jira Miscellaneous Workflow Extensions (JMWE)</u>.

See step by step instructions in my <u>verify approval in a Jira workflow</u> article in The Hub.

Service request automation

Sometimes support efficiency and effectiveness is measured by the effort or time required to complete certain tasks. Wouldn't you like to eliminate manual tasks so you can focus on solving problems instead? Building automation rules is useful and fun! Here are some ways to automate service request work and get to a solution faster.

Idea	Jira Service Management Features
Assign new requests to the correct resource, based on information submitted.	 Use the built-in component routing feature to assign issues to the designated component lead. Use the built-in automation feature in JSM Cloud, or a marketplace app in JSM Server and Data Center, to change the request's assignee based on desired criteria.
Create issues in the facilities and security team's Jira projects when an onboarding or offboarding request is created.	 Use an automation rule or a workflow or scripting app to automatically create issues triggered by other issues. Tip: Also copy important information from the original request to the new issues. This way no one has to search somewhere else for existing information.
Allow a request to skip approval if the cost is less than \$100.	 Create custom number field to collect estimated cost information. Then use an automation rule or an app to check the cost and skip forward in the workflow when applicable.

What other ways can you reduce manual work at your organization?

JMWE

Did you know that <u>Jira Misc Workflow Extensions</u> ("JMWE") automates more than just workflows? In addition to its workflow conditions, validators, and post functions, JMWE also automates system events and schedules actions.

Challenge

The leadership team knows incident reports provide valuable insight, but agents are busy and don't always remember to create them. When users forget something important, use Jira to help remind them.

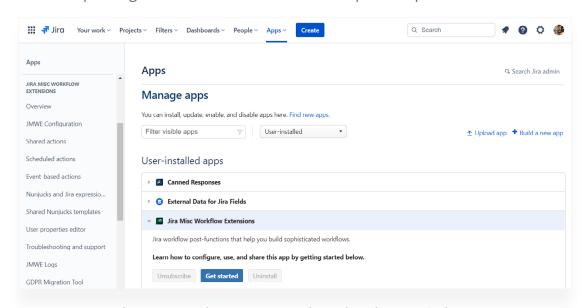
Note: JSM has an incident report feature, which collects event data in the form of a new Jira issue. I prefer to record event data in another application, like Confluence. For this example, we'll direct agents to wherever location important records are stored.

Solution

Create an event-based action to remind agents to create incident reports.

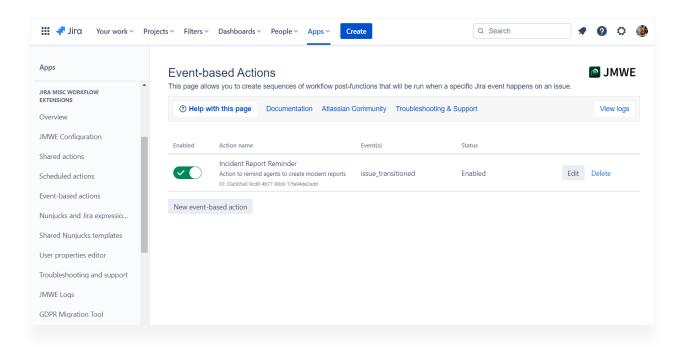
Constraints

- Include agents using a specific Jira Service Management project
- · Create notifications only for high priority incidents
- Send the reminder after an incident is closed
- Suppress notifications for duplicate events
- Help the agent find the standard incident report template



The "Apps" admin page showing JMWE's admin options in the left sidebar

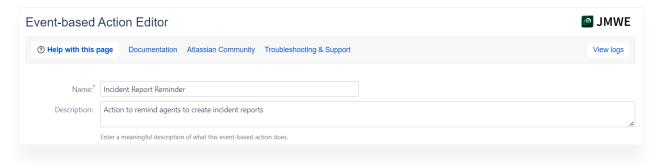
- Login to Jira as an application administrator.
- Click the "cog" or "gear" icon at the top right and select the "Apps" admin page.
- In the left sidebar, click the "Event-based actions" link under the "Jira Misc Workflow Extensions" section header.



The JMWE Event-based Actions page

On the "Event-based Actions" page, click the "New event-based action" button.

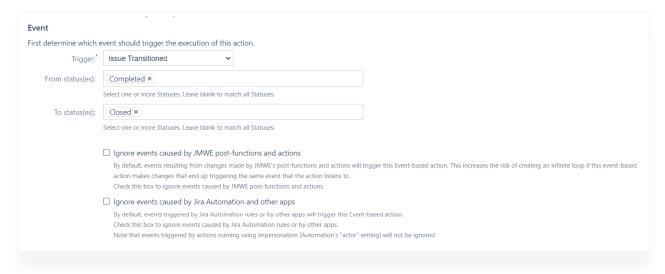
EVENT-BASED ACTION



General action settings

Next, configure the following settings:

- · Name: Incident Report Reminder
- · Description: Action to remind agents to create incident reports



Event settings

The event should run only when an incident transitions from the "Completed" status to the "Closed" status. I chose these statuses so incidents are fully addressed before incident details are requested. When there's a true emergency, solve the problem first and follow-up with documentation and procedural requirements later.

- · Trigger: Issue Transitioned
- From status(es): Completed
- To status(es): Closed

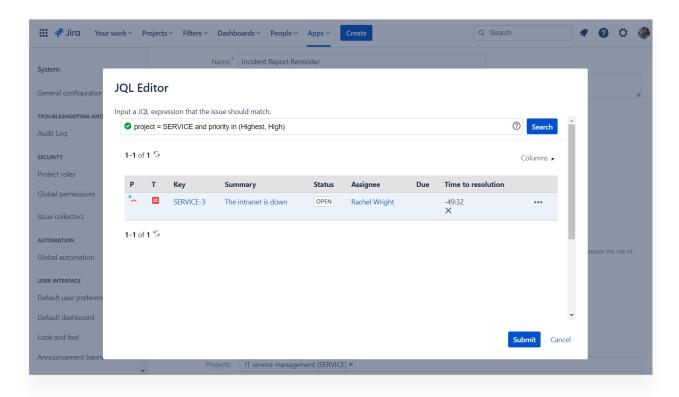
Scope

Note: There are many ways to limit event scope. Use the "Projects" and "Issue types" selection fields, specify a specific project with a JQL query, or use the conditional execution feature.

All methods are shown below as an example of the possibilities. Some settings are redundant.

Limit the scope to high priority incidents in the SERVICE project. (Low priority issues likely don't warrant an incident report.)

- Projects: IT service management (SERVICE)
- · Issue types: [System] Incident
- · Enable the "Only apply to issues that match a JQL filter" setting



JQL query and result

JQL expression:

1project = SERVICE and priority in (Highest, High)



Always validate JQL query syntax and sanity check the issues returned before using a query in an app. If the app doesn't have a validation function, test the query using Jira's "Advanced issue search" function.

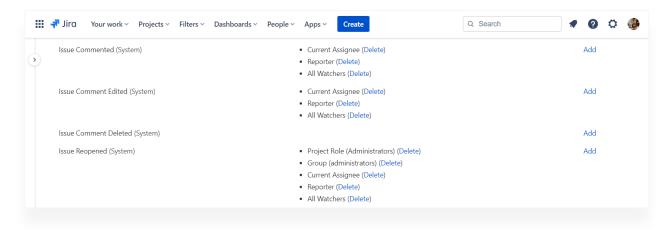
POST-FUNCTIONS

Now configure specific settings for the reminder message.



Click the "Add post function" button to add one or more functions

In this section, click the "Add post function" button. In the overlay, select "Comment issue(s) (JMWE app)."



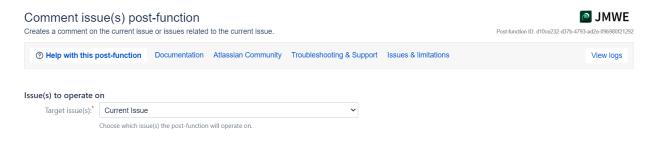
Current assignees receive the "Issue Commented" notification

I selected this post function intending to combine it with standard Jira notification functionality. Assuming the project's notification scheme is configured as shown, Jira will already send an email notification to the assignee when a comment is added. As such, there's no need to build additional or custom email functionality.

The "Comment issue(s) (JMWE app)" post function has its own settings. Configure the following:

ISSUE(S) TO OPERATE ON

Next, specify which issues are impacted. There are many options like: subtasks of the current issue, issues in an epic, linked issues, JQL segmentation, and more.



Post function settings

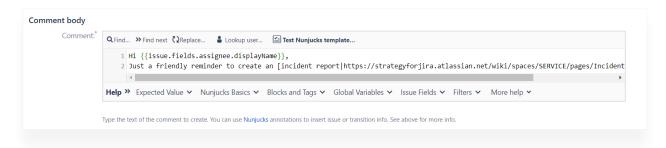
Target issue(s): Current Issue

COMMENT BODY

Next, craft the message for the comment field. The example contains the assignee's name as a variable and a hyperlink formatted with wiki markup.



Use the "Test Nunjucks template" button (pictured) to test message output. Read more about Nunjucks expressions at: <u>How to insert information using Nunjucks annotations</u>

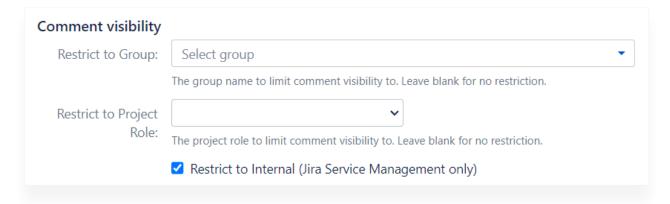


Post function comment

Hi {{issue.fields.assignee.displayName}},
Just a friendly reminder to create an [incident report |
https://strategyforjira.atlassian.net/wiki/spaces/SERVICE/pages/Incident+Reports]
for this event!

COMMENT VISIBILITY

Use Jira's comment restriction capabilities to control visibility. For this example, prevent external customers from receiving notifications with the "Restrict to Internal" option.

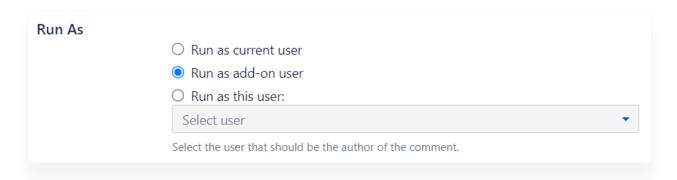


Additional post function comment settings

• Enable the "Restrict to Internal (Jira Service Management only)" setting

RUN AS

Choose which user is listed as the rule's executor, sometimes called the "actor." By selecting the "Run as add-on user" option, the message is posted as the "user" called "Jira Misc Workflows Extensions." You can also leverage an existing service account.

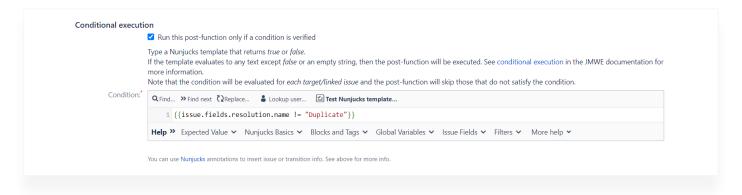


Post function actor settings

• Run as: Run as add-on user

CONDITIONAL EXECUTION

Here's another way to limit scope. Enter a simple expression to exclude issues identified as duplicates of other incidents. Use Jira's resolution field to exclude incidents that can't be reproduced, are known issues, or any other scenario that doesn't need a report.



Post function conditions

- Conditional execution: Enable the "Run this post-function only if a condition is verified" setting
- Condition: 1{{issue.fields.resolution.name != "Duplicate"}}

Finally, save the post function and save the event.

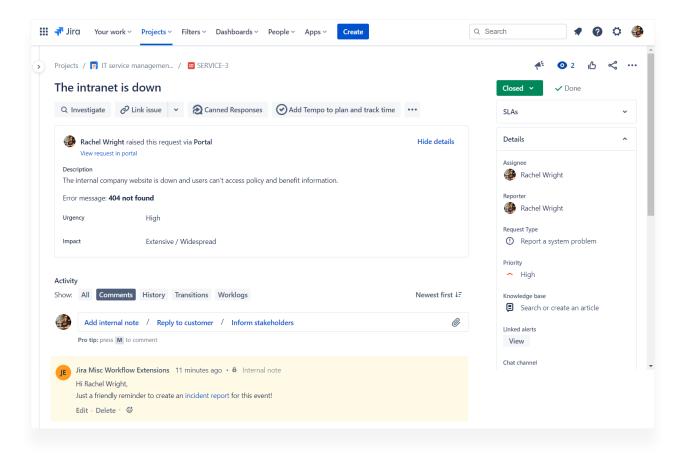
TESTING

To test the event, create a new request in the SERVICE project of type "Incident." Set the priority to "Highest" or "High." Transition the issue through the workflow and select a resolution of "Done" (or anything except "Duplicate"). Transition the issue to its final status. Refresh the issue and look for the new comment at the bottom of the page.

Q Expert tip!

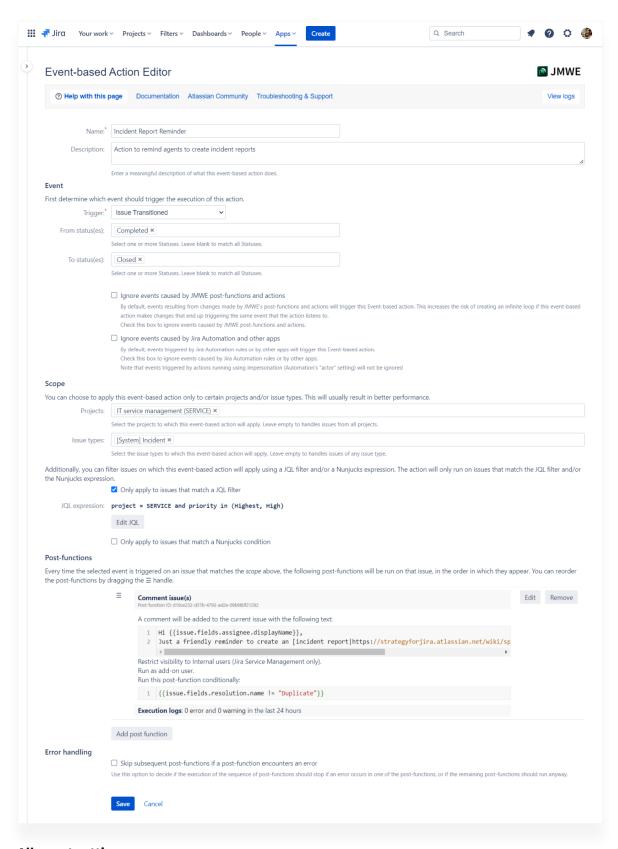
Don't forget to test negative conditions too! For example, check that a comment is not added when the issue is a service request, the priority is low, and the resolution is "Duplicate."

Results



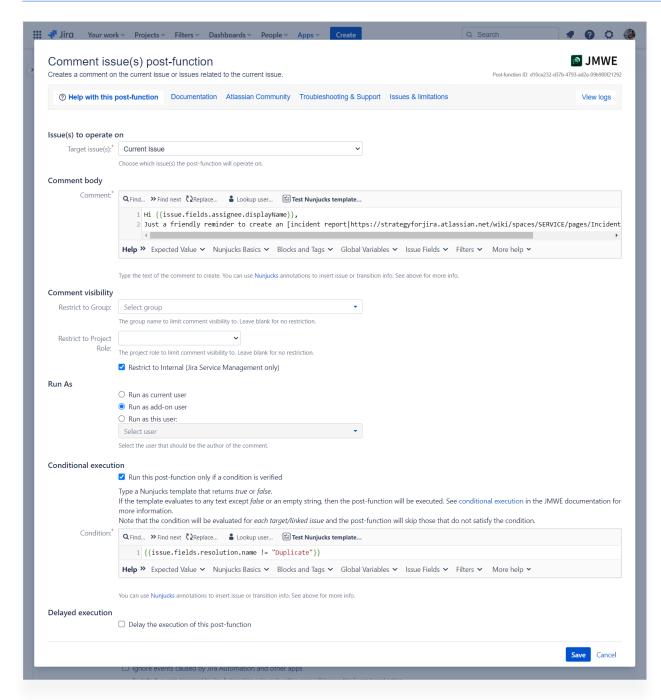
Comment automatically added when the issue is closed

Here's the final result. The issue is in the "Done" status and a reminder is present in the comments area at the bottom. The reminder (highlighted in yellow) shows it was posted by the "user" called "Jira Misc Workflows Extensions." Comment visibility is restricted to internal users. Finally, the name in the comment body is dynamic and the link takes the user to where incident reports are stored.



All event settings

This image shows all event and post function settings on one screen.



Close-up of post function settings

Here's a close-up of the settings for the post function section.

Q Expert tip!

You can also use the "Comment issue(s) (JMWE app)" post function in workflow transitions.

Q Expert tip!

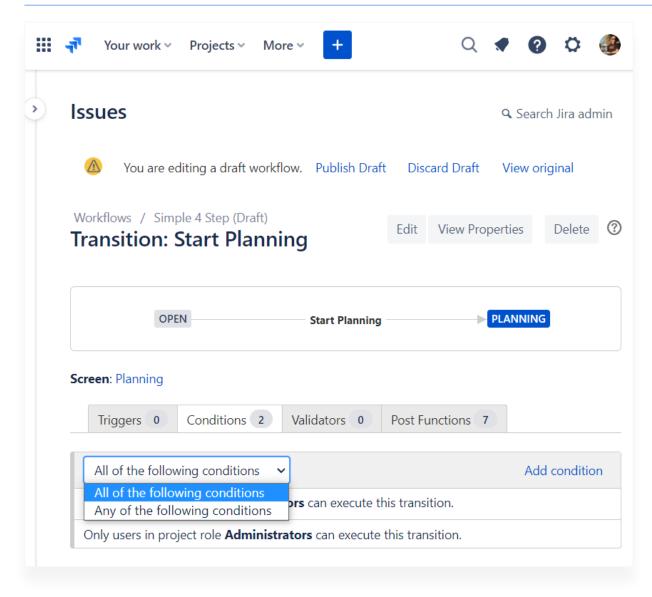
At one company, agent performance was measured by time to completion, regardless of the circumstances. By measuring success this way, team members were more focused on closing requests quickly than providing excellent support. Yes, metrics play an important role in the overall support process, but don't rely too heavily on any one goal or measurement.

Approval workflow best practices

- Provide positive and negative options for workflow approval steps (e.g., an "approve" transition and a "decline" transition)
- For highly regulated industries, consider automatically logging the name of the approver and the approval date
- Determine whether or not an approval needs to be restricted to certain users
- If restriction is needed, build contingency options to handle potential problems, like when the designated approver is unavailable
- Use roles and groups instead of individual usernames in schemes and workflow behaviors
- Don't create unnecessary workflow bottlenecks

Q Expert tip!

When adding multiple transition conditions, be sure to consider whether any or all conditions must be true. The option is shown in the image.



Workflow condition option

S Expert tip!

What status does an issue go to if approval is declined? If the answer is the final "Done" or "Closed" workflow status, use a "decline" transition to automate standard issue completion steps. For example, make closing the issue easy by setting a resolution in the background with a post function. A resolution of "Won't Do" (or similar) explains why the issue is in its final status. There's no need to launch a transition screen for the user to manually select a resolution.

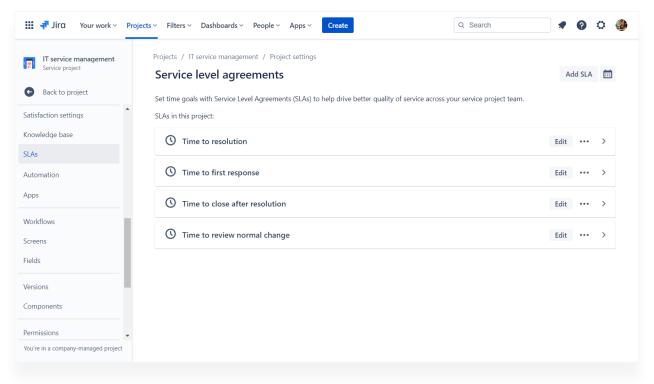
SLAs and continuous improvement

A service desk is a dynamic set of processes. You'll want to continuously report on results and improve processes as you go. One useful ITSM metric is adherence to service level agreements (SLAs). SLAs measure how closely support goals align with customer expectations. For example, "Did we respond to customer requests before the deadline mentioned in our contract?" or "Is our vendor delivering the needed service within the expected time window?" JSM provides an easy way to measure whether goals are met for individual requests or many requests over time.

The <u>Time to SLA app</u> is an awesome way to track SLAs with both external customers and internal teams. This app is available for both Jira Service Management and Jira Software, so IT and development teams can meet their respective deadlines as well as collaborate with support and service desk teams.

JSM comes loaded with some sample SLAs to get you started. Let's take a look.

Built-in SLAs



Built-in JSM SLAs

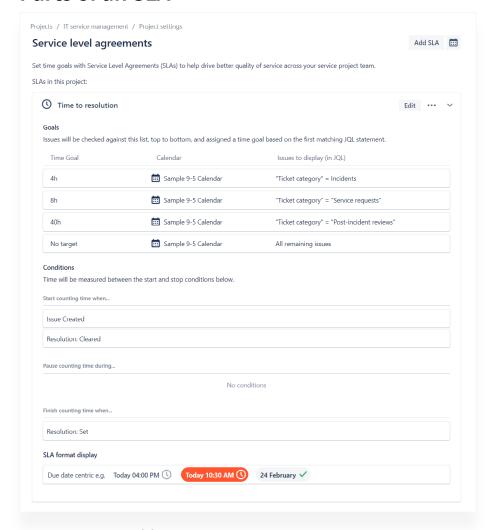
JSM service projects come with multiple example SLAs including:

- Time to resolution
- · Time to first response
- · Time to close after resolution
- Time to review normal change

The built-in SLAs contain sample settings to segment requests, determine conditions, and set goals. The examples provide a great starting point, but you may need to modify the default settings or create new custom SLAs for your organization's support policies.

Project administrators and application administrators can create and manage SLA settings. Service projects may have zero, one, or many SLAs.

Parts of an SLA



SLA goals and conditions

Each SLA contains the following parts:

- · A metric to track, like "Time to resolution" in the screenshot
- One or more time goals, like four hours for incidents and eight hours for service requests
- · A calendar to specify work hours
- A JQL query to segment issues
- Conditions to define when to start counting time, pause the counter, or stop the counter

PLACES TO VIEW SLAS

There are many places to see SLA metrics depending on your deployment type and permissions.

AGENTS

Agents can see SLA goals in the following places:

- On a filter or search results page
- In a default or custom queue
- In an individual support issue
- In reports

PROJECT AND APPLICATION ADMINS

Project administrators and application administrators can also create and edit SLAs in the project admin area.

JIRA SOFTWARE AND JIRA WORK MANAGEMENT USERS

Only licensed agents can see SLA goals in JSM.

PORTAL CUSTOMERS

In Cloud, you can use an app from the Atlassian Marketplace to show SLA information to portal customers. In Data Center (version 4.18.1 and later) there's an option on the SLA creation screen to share the SLA's status and time metric with customers.

SLA Reports

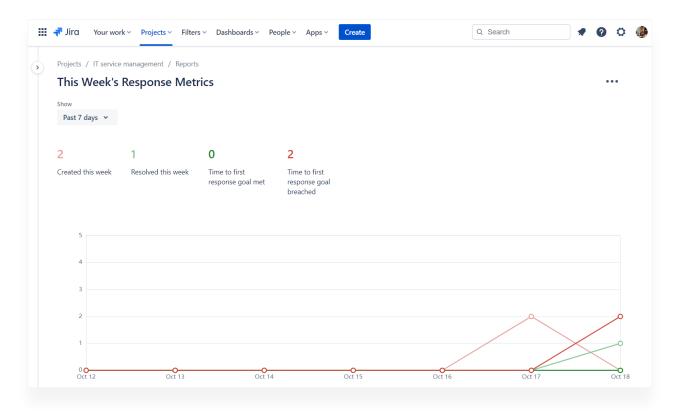
Service management projects have additional built-in reports to help determine goal adherence. Look for them on the "reports" page accessible from a project's left sidebar.

The built-in reports include:

- · Time to resolution
- · SLA met vs breached
- SLA success rate
- SLA goals (in Server and Data Center)

CUSTOM REPORTS

You can also create custom reports. Look for the "New report" link at the bottom of the project's "reports" page sidebar.



Example custom report

The example custom report compares issue activity with the "Time to first response" SLA. The report scope is restricted to the past seven days. There are four data sets with custom names, colors, and further segmentation using JQL. The report shows that of the two issues created this week, only one is resolved, and neither issue met its "Time to first response" goal.

Interested in creating more personalized SLA reporting? <u>Time to SLA</u> offers easy-to-use built-in reports and gadgets that help to create personalized reports with your own Jira filters using JQL functions.

SLA Automation

JSM Cloud has built-in automation to help the support team manage and monitor SLA activity. Here's one easy example:

Problem

The team is focused on strategic company priorities and aren't resolving requests within the agreed timeframe

Solution

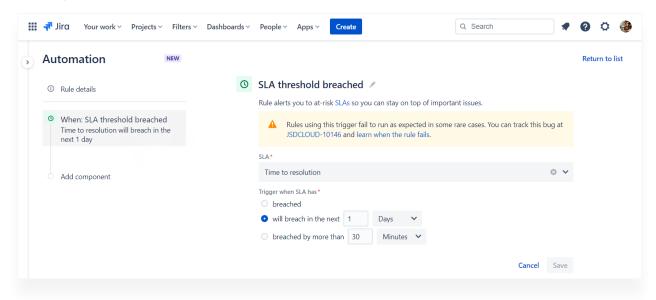
Automatically remind the team of requests approaching their deadline

Implementation

Send an alert one day before the "time to resolution" deadline. Take advantage of built-in comment functionality like emailing mentioned users and converting issue keys to show additional issue data. (Shown in the "result" screenshot below.)

- · Automation Trigger: SLA threshold breached
- Automation Action: Comment on issue
 - Other available actions: Send Slack message, Send Microsoft Teams message, Send email, and Send Twilio notification

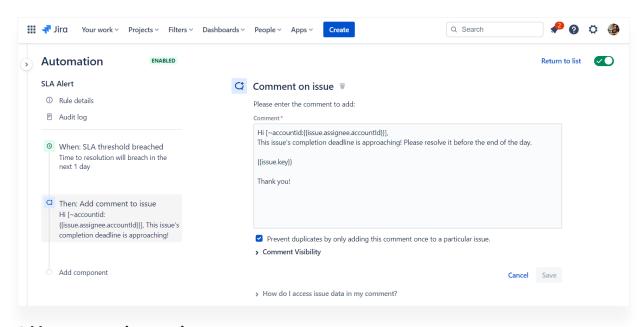
Configuration



SLA threshold breached trigger

Add a trigger and select the following settings:

- Trigger: SLA threshold breached
- SLA name: Time to resolution
- · Trigger when: will breach in the next 1 day



Add comment to issue action

- Add an action and select the following settings:
 - ° Action: Add comment to issue
 - ° Comment copy: See below
 - ° Comment visibility: Service Management projects Internal only

Comment Copy

Hi [~accountid:{{issue.assignee.accountId}}],

This issue's completion deadline is approaching! Please resolve it before the end of the day.

{{issue.key}}

Thank you!

Notes:

Use "[]" (square brackets) and the "~" (tilde) symbol to mention the assignee and generate a Jira email.

Use "{ }" (curly brackets) with <u>smart values</u> to dynamically display the issue's key. When an issue key is included in a comment, Jira will automatically display the key, summary, and status. The issue is also hyperlinked.

Result



Automation for Jira 19 seconds ago

Hi @Rachel Wright),

This issue's completion deadline is approaching! Please resolve it before the end of the day.

DEV-2: As a customer, I'd like a better description of the "Custom Training" service TO DO

Thank you!

Automatically created comment

The comment is automatically added to the issue, is only visible internally, the current assignee is tagged, and the issue details are included.

Troubleshooting

Sometimes SLA results show inconsistent data or unexpected results. When this happens, here are some things to try.

Wait and recalculate

Seeing strange results? Try editing an SLA and saving it to automatically trigger a recalculation.

SLA changes are not applied immediately. JSM needs to perform a recalculation and re-index process first. The length of time a recalculation takes depends on many factors, like how many issues are impacted. SLA times for impacted issues and related reports are often inaccurate during this time.

Recalculations occur for any unresolved issues.

Simplify queries and settings

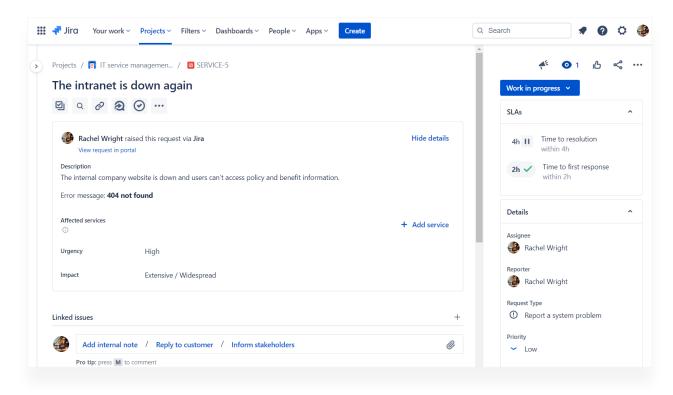
Too many SLAs, conditions, or inefficient JQL queries can cause timeouts. Always look for ways to simplify queries and limit their scope as much as possible.

Examples

Original JQL	Improved JQL	Notes
Assignee = currentUser()	 ProjectCategory = X and assignee = currentUser() Project in (X,Y,Z) and assignee = currentUser() Project = X and assignee = currentUser() 	Limit the scope to specific projects
Project = Y and assignee = currentUser()	 Project = Y and assignee = currentuser() and statusCategory = Done 	Remove completed issues to reduce count

Check for goal changes

Also check for any recent goal configuration changes. Changes during peak use hours are likely to impact issues being worked.



Four-hour time to resolution goal

For example, this incident's resolution goal is four hours. Two hours have elapsed meaning two hours remain. If the SLA's time goal is lowered to three hours, then only one hour remains. Agents may not be able to meet the new deadline, issues are more likely to miss the goal, or issues may breach immediately.

If your organization is expanding or has complex SLA needs, try out the Time to SLA app. Time to SLA expands capabilities with additional JQL functions, custom reports and dashboards, additional start and stop triggers, and a color coded information panel to help the team stay on track. There's an issue countdown timer and custom automatic notifications so deadlines are met. There are additional calendar features to accommodate time zones and custom durations like half and full day availability. The additional features in the Time to SLA app ensures your organization can manage numerous SLAs and varying contractual obligations no matter the complexity or scale.



Aappfire

How do I manage Service Level Agreements (SLAs)





Preseted by Kerry O'Shea Gorgone with guest Onder Ozcan

Want to learn more about SLAs? <u>Check out this conversation</u> between Appfire's Onder Ozcan and Kerry O'Shea Gorgone, where they discuss how to manage Service Level Agreements in a helpful 10 minute video.

JSM reporting

In addition to Jira's built-in filtering and reporting features, JSM adds additional ways to view request data and progress. There are special reports to view workload per agent, customer satisfaction, and other support-specific metrics.

Reporting methods

There are multiple ways to assess data in JSM. The methods include:

- · Configuring standard and custom reports
- · Creating custom filters and queues
- Building custom dashboards to display statistics
- Extending JSM with apps from the Atlassian Marketplace
- Developing custom reporting functionality using the REST API
- Exporting request data to separate business intelligence applications

Built-in reports

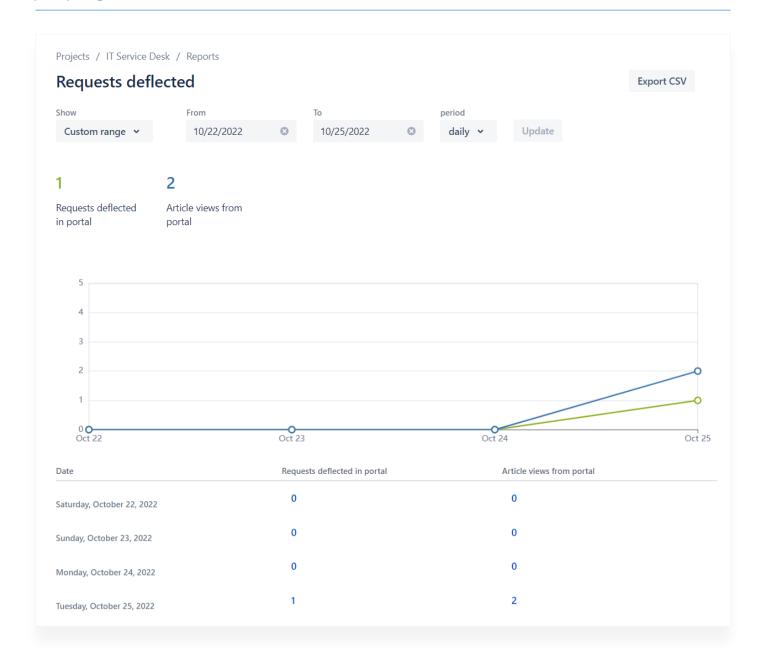
Jira comes with many built-in reports for viewing information. Jira Work Management contains a standard set of reports for general analysis and planning. Jira Software has all the reports available in Jira Work Management, plus additional development-specific views. Jira Service Management has its own unique set of reports plus the ability to create custom reports. JSM reports expose how many requests are assigned to each agent, a team's satisfaction rating, how users leverage self-service content, and more.

Here's a list of reports available in the most popular Jira application types.

Work Management Project Reports	Software Project Reports	Service Management Project Reports
Issue Analysis Average age report Created vs. resolved issues report Pie chart report Recently created issues report Resolution time report Single level group by report Time since issues report Forcast Management Time tracking report User Workload report Version workload report Other Workload Pie chart report	All reports in Jira Work Management plus: Agile Burndown chart Burnup chart Sprint report Velocity chart Cumulative flow diagram Version report Epic report Control chart Epic burndown Release burndown Cycle time report Deployment frequency report	A unique set of reports including: • Workload • Satisfaction • Requests deflected • Requests solved • Average time spent • Created requests • Created vs resolves • SLA met vs breached • Incidents by priority • Change by type There's also a custom report feature

EXAMPLE REPORT

My favorite JSM report is "Requests deflected". It measures the effectiveness of knowledgebase and self-service content. Well-written help content increases the chances of customers solving their own problems. This report shows how many times a customer solved their own problem and how many knowledgebase articles were viewed.



JSM "Requests deflected" report

In the example report, on the same day, one request was deflected and two articles were viewed.



I find the count labeled "Requests deflected in portal" a little misleading. I thought JSM was calculating the number of times a request was started but abandoned. Instead, it means a customer liked a self-service help article.



One question survey in self-service article

The question "Does this article help?" appears at the bottom of every article. If the customer clicks "Yes", the "deflected" count increases.

To see a count of views per help article, click the "Knowledge base" option in a support project's left sidebar.

Custom reports

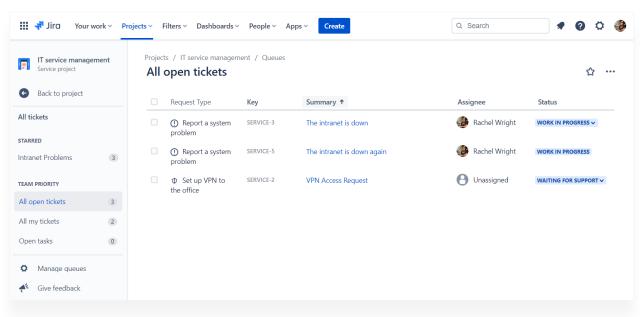
In JSM, you can also create your own custom reports to track specific goals. For example:

- How many requests were resolved this week?
- How many times was the company intranet down this quarter?
- Were any VIP or high priority requests completed today?
- · And more

To create a custom report, click the "New report" option at the bottom of a support project's left sidebar.

If you need additional reporting and data analysis features, there are hundreds of apps to explore in the Atlassian marketplace, like the <u>Enhancer Plugin for Jira</u> app to boost reporting capabilities.

Queues



Standard and custom queues

In addition to filters, Jira Service Management also has queues. A queue is a shared view of issues. A queue uses JQL (Jira Query Language) to limit the number of issues returned in a search. It works just like a filter except it's stored with a support project, not in a particular user's profile.

Queues are project-specific and intended to help each support team segment and categorize requests. You can create a queue for any data that's queryable in a standard or custom field. Additionally, queues are not mutually exclusive; a request can appear in multiple queues at the same time.

Click the "Queues" link in a service project's left sidebar to access, create, or modify queues.

Access and permissions

Access to support-specific features requires a Jira Service Management agent license, which is different than a Jira Software or Work Management license.

TO ACCESS QUEUES

You'll agent permissions to access queues and either project administrator or application administrator permissions to create or edit them.

TO ACCESS REPORTS

To see report data, you simply need access to the support project. You'll find the "Reports" link in a project's left sidebar.

TO COLLABORATE WITH AGENTS

Licensed Jira Software or Jira Work Management users are known as "collaborators" in JSM. Collaborators can view and add issues, comments, and attachments.

They don't have access to queues, reports, or SLAs, however. Collaborators also can't log work or transition issues. They have project access to provide assistance and context to the support team, but not actually fulfill the support request.

Customer feedback

Marking a request as resolved does not mean the process is complete (we *are* providing customer service after all). The only way to improve is to measure results, collect feedback from others, make small adjustments over time, and continuously repeat the process.

How do you improve if you don't know what your customer's really thinking? Achieving customer service excellence is a journey that never ends. Your process will grow and change as your organization matures and customer needs evolve.

Identifying a Strategy

Don't go in blind. Customer feedback is complex and can offer a lot more than just answers to your questions. Before you start writing survey questions, it's important to understand the following:

WHAT IS THE GOAL OF COLLECTING THIS FEEDBACK?

I always start survey writing (and Jira project configuration) by thinking about the end goal. What information would you like to report on? What data must be present to produce a meaningful report?

For example, in Jira, to know how quickly a problem was solved, make sure specific time and date information is available. To know how many users received new keyboards this year, you'll need a way to distinguish those requests from other equipment requests. In Jira, you can segment based on a component, custom field, label, or any other flag or unique identifier.

For feedback surveys, think about what you want to measure and how you'll obtain that information.

WHO IS THE CUSTOMER?

As previously discussed, understanding your customer is vital to this process. If you're on an IT team, you're likely supporting internal employees with technical requests. If you're on a sales team, your customer is external contacts and leads. If you're on an HR team, you likely have both employees ("internal customers") and job candidates ("external customers").

Whether the customer is outside the organization, inside the organization, technical or non-technical, etc., are factors that determine the best way to collect feedback from that audience. Make sure the customer is properly understood, defined, and segmented (if applicable) before creating surveys and questions.

WHAT DOES THE CUSTOMER NEED, EXPECT, OR WANT?

Next, imagine you're the customer. What types of help, answers, or items does the customer normally ask for? Is there a gap between what you provide and what the customer needs or expects? This information might determine how best to ask a question.

WHAT WILL YOU DO WITH THE FEEDBACK?

Once you have the feedback, what will you do with it? This answer may determine how you collect, store, share, or act on the information.

The worst thing you can do is leave the customer wondering whether their feedback was received, whether it will be acted upon, or whether you're even listening. Treat surveys as an opportunity for a conversation. The conversation should never be one-sided. Imagine how great the customer would feel if you contacted them to learn more about one of their responses? That would certainly make me feel like my input was received, understood, and appreciated.

There are plenty of tools to help the customer know their feedback was received, thank the customer, and follow-up once you take action based on the feedback (or even if you don't). Make sure feedback doesn't perish in a metal box or an email account that no one checks. Surveys for Jira ensures that customer feedback is tailored for your specific needs and audience. Collecting insight and metric reports that are always on point, providing effective and reliable data, everytime.

HOW DO YOU DEFINE SUCCESS?

Finally, what metrics will you use to determine whether you're meeting your goals? Is a particular response considered positive, negative, or neutral? How will you "score" the results to derive sentiment?

For example, will a human read free text feedback and decide whether the customer is happy or not? Or, if you use a rating system, what do the numbers or selections represent? For example, does receiving a response of "3" on a "1-5" scale mean needs are met or not?

Determine these answers early so you can consider that information when crafting questions and be ready to analyze the responses.

Do your research

It's important to know your audience and design surveys that users will actually complete! Think about the one piece of information that you're really trying to gather, and tailor your survey for that purpose. Listen in on this conversation between me and Appfire's Kerry O'Shea Gorgone: "What should I do before I build the customer feedback form?"



Feedback requirements

To collect and process feedback in a meaningful way, you'll need the following elements:

- A well-defined set of questions, objectives, or goals. For example, "Do customers like the new chicken wings at our restaurant?" or "Why did the restaurant have fewer customers in March?"
- A group of targeted and willing participants
- A mechanism to collect the data (e.g. paper forms, emails software, and other tools)
- A way to measure the data and identify frequent patterns
- · A commitment to take action based on the results

Types of Feedback

CSAT: Customer Satisfaction Score ("CSAT") is a short, simple format that measures product or service satisfaction, often using a numerical rating scale.

NPS: Net Promoter Score ("NPS") measures overall customer loyalty instead of a specific product or service. An NPS survey often asks one broad or overarching question.

VOC: Voice of the Customer ("VOC") focuses on the customer's most important needs and expectations and measures individual data instead of aggregate data. VOC surveys contain open-ended questions that customers can answer in a variety of ways.

Bonus! Want to amp up your design of useful surveys and questions? Add this book to your reading list, "Perfect Phrases for Writing Employee Surveys: Hundreds of Ready-to-Use Phrases to Help You Create Surveys Your Employees Answer Honestly" by John Kador and Katherine Armstrong.

Tools for collecting customer feedback

User feedback is one way to measure team success. As a leader or a conscientious team member, there's no better way to understand the customer's experience than directly asking for information after a request is fulfilled. Collecting this vital internal or external information doesn't need to be complicated or expensive, though. Here are some simple ways to get started collecting and analyzing customer satisfaction data.



∆appfire

How do I use tools to collect and manage customer feedback?



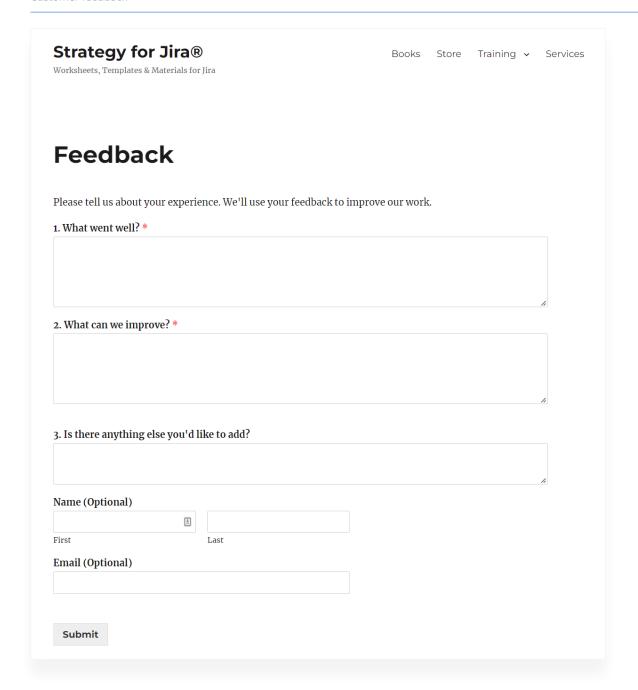


Presetted by Kerry O'Shea Gorgone with guest Rachel Wright

<u>Check out this helpful 10-minute video</u> where I chat with Appfire's Kerry O'Shea Gorgone about feedback management tools for small teams, big teams, and every size team in between!

UNSTRUCTURED DATA

Unstructured data means information not stored in a standard format. For example, a text answer to an open-ended question or a VOC response. Here are some ways to collect this type of information.



Simple Website Feedback Form

WEB

- Add a feedback or contact form to your website
- Turn on your blog's "comments" feature or your online store's "reviews" functionality
- Use a web-based survey tool to host questions and collect responses

Many content management systems include these abilities, and you can easily add them with plugins and extensions if your CMS doesn't come with the functionality built in.

EMAIL

- Automatically or manually send a feedback request after a payment
- Use other business tools you already have to send emails (E.g. your customer relationship management, payroll, or email marketing software)

CONFLUENCE

- Create a Confluence space or page to collect responses
- Use a survey app from the Atlassian Marketplace. (There are more than 50!)

Featured App: Easy Forms for Confluence

My favorite Confluence survey app for Data Center is Easy Forms for Confluence. It's a fast way to build a form and collect feedback of all kinds. Use it to conduct an employee satisfaction survey, post a poll to determine where to host the next company anniversary celebration, collect feedback after an incident, and much more.

The app has a drag and drop interface, and no coding or technical skills are required. Just add the Easy Form macro to a Confluence page, add form questions, and publish the page. After feedback is collected, you can view the results right in Confluence with the Easy Form Report macro or export the results in json, xml, xls, or csv format.







Place the form using the Easy Form Macro

End user's view of the form

On page form results with the Easy Form Report macro



Results exported in spreadsheet format

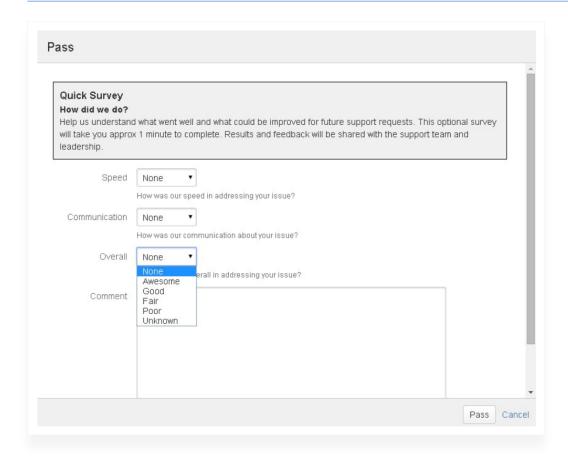
STRUCTURED DATA

Structured data has a predefined format or model which makes it easy to calculate, compare, or analyze. For example, a consistent rating scale allows multiple responses to be easily tabulated and charted.

JIRA

Jira's structure of standard and custom fields makes its data easily searchable and sortable.

One of the easiest ways to get and track user feedback is to collect it in the Jira issue containing the request. To do this, I set up a short, optional survey as part of the final step of the workflow. I ask each user to verify their request was addressed with "Pass" or "Fail" workflow transitions. When "Pass" was selected, a transition screen launched to show optional survey questions.



A simple workflow transition screen

Sample Questions

- 1. How was our speed in addressing your issue?
- 2. How was our communication about your issue?
- 3. How did we do overall in addressing your issue?

How To

- Create one new custom field, of type "Select list (single choice)," for each question to ask.
 - Add simple response selections like: Awesome, Good, Fair, Poor, and Unknown
- Create one new transition screen and add the custom fields created in the previous step.
 - Note: The "Comments" field will be present on the screen by default. There's no need to create another field to gather feedback in text format.
- In the final workflow transition, show the screen created in the previous step.

Create a dashboard to count and show the results.

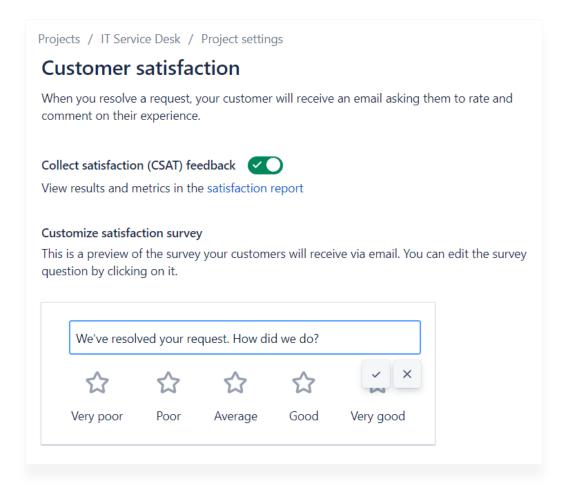
Additionally, you could simply ask users to add their feedback to the standard comments field or use a survey app from the Atlassian Marketplace.

One benefit of this approach is the customer and all their feedback stays in one application, so it's automatically associated with related issue data.

JIRA SERVICE MANAGEMENT

In JSM, a basic CSAT survey is built in. There's not much to customize, but it's a quick and easy way for customers to provide feedback and for leadership to measure user sentiment.

This feature is available for any "service management" Jira projects. In the project, click "Project settings" in the left sidebar to go to the project admin area. Then click "Satisfaction settings" in the left sidebar.

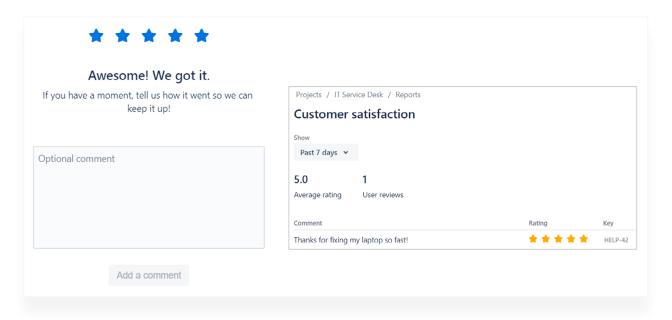


Customer satisfaction survey settings in JSM

On this page, you can turn the feature on or off or customize the question copy. Click on the current question to edit it.

How It Works

- When a request is resolved, a quick, one-question survey is emailed to the customer.
 - ° The customer can select their response directly in the email message.
- After the customer clicks a response, a browser window opens to collect any additional feedback.
 - If a comment is submitted, it appears in a dedicated, project-specific report.
 - ° None of the survey data displays in the Jira issue or JSM request.



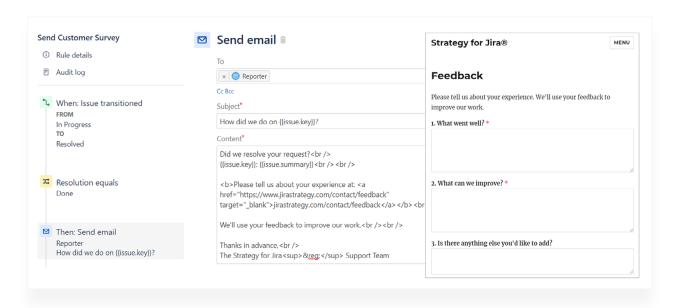
A five star review and a satisfaction report

AUTOMATION

I try to automate manual tasks every chance I get. Here's an example of automating survey delivery and a response.

EMAIL CUSTOMERS WITH SURVEY LINK

What if you already have a different survey tool or don't want to use the one in JSM? You can use automation to send the customer a link to wherever satisfaction data is collected.



A simple automation to email a link to a survey

In the example, the automation is activated when an issue transitions from the "In Progress" status to the "Resolved" status and the Resolution is "Done." An email is sent to the reporter, asking them to click a link, to provide their feedback. In this example, I simply directed users to my website's existing feedback form.

EMAIL A RESPONSE

Another way to use automation is to create a response system. You can use this to thank the customer for their feedback, conditionally ask for additional details, or notify a customer that their suggestion was implemented.

Here are some ideas:

- When a Jira issue is completed, automatically send a "thank you" email to the reporter.
- Automatically send survey messages based on issue data. For example, send different messages if the reporter is a VIP, if the request is for a high cost item, or if the response was slower than desired. You can segment messages by any data available in the Jira issue. Then simply add conditions to the automation to control which messages are sent when and with what content.
- If a CSAT or NPS score is low, automatically send a message requesting a chat, meeting, or additional information.

 Tag similar bugs, requests, or issues with a unique label. Then use the label to automatically send a pre-written message with a progress update. (Just make sure not to send duplicate notifications to the same customer!)

There are also plenty of survey related apps in the Atlassian Marketplace. Search for "survey" and limit the results list to the JSM application.

REAL-TIME DATA

What if you'd like to collect more up to date information? There are many ways to collect immediate stats and feedback data. For example, add an SLA app to Jira, or use the built-in SLA functionality in JSM to monitor whether responsiveness goals are being met. Or, use a stand-alone, website, or Jira chat application to collect real-time information. It's easy to automatically turn that information into Jira issues with an app from the Atlassian Marketplace.

EXPERIENCE MANAGEMENT

Finally, if you don't have an internal team to manage information collection and analyze results, hire an employee or customer experience management company to do it for you.

PROS

- Fully outsource survey delivery, information collection, and analysis
- A dedicated resource to manage information no matter when it's received

CONS

- Less control over information collection and analysis methodology
- Potentially expensive

Additionally, customers may be more (or less) likely to respond, or provide more candid responses to a third-party.

Customer feedback best practices

In this video, I put on my former web development hat to share form design tips and best practices. How long should a feedback form be? How can conditional logic reduce form abandonment? How can you make language in Jira Service Management more customer friendly? The answers are in this 10-minute video: "How do I design user friendly support forms?"





Need to reply to customer feedback at scale?

In <u>this 10-minute video</u>, Appfire's Kerry O'Shea Gorgone and Rachel Wright discuss all the best ways to do it!



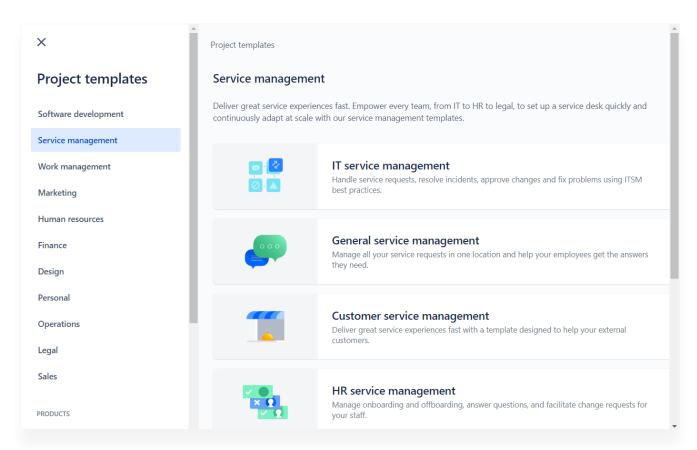
JSM project templates

Choosing a template is the first step in configuring a support project. JSM offers a wide array of project templates to suit your organization's support needs. Templates provide a simple, effective, and streamlined way to do your job the right way from the start.

Project templates provide an initial configuration structure containing common key elements. For example, an IT service management project template creates a support-type project with issue types commonly used by support teams. (e.g., Change, incident, etc.) The sales project template however, creates a work management-type project featuring custom statuses like "contract sent" and "negotiation." The IT team and the sales team have very different roles and procedures, and their Jira project configuration should reflect that. The templates help administrators create new projects quickly while retaining the ability to further customize many elements of the configuration.

Cloud Templates

Each Jira application and deployment type comes with different templates. Jira Service Management templates contain settings specifically designed for support functions. They're commonly used by teams like IT, customer service, and facilities, but any team that provides service or support can use them.



JSM Cloud project templates

The screenshot shows project templates in Jira Service Management Cloud. Additional templates for other uses are categorized in the left sidebar. There are many project templates to choose from, so pick one that most closely represents how the project is used and who uses it.

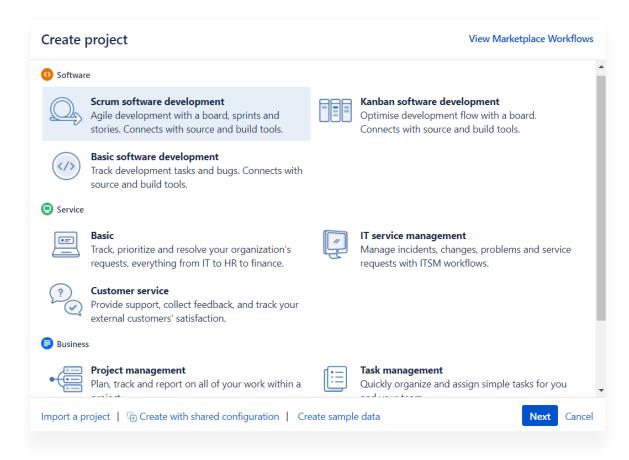
When I'm evaluating a template, the first thing I do is review who it's for and what it contains.

For example, the IT service management template says it's built specifically for IT and DevOps teams. The template information page lists the request types and other elements included in the template. Some templates also include custom workflow steps and other settings.

Q Expert tip!

I often base my template choice on how similar or different the default workflow is to what the team needs. Workflow building is the biggest configuration step, so starting with similar statuses and transitions saves time.

Server and Data Center templates



Server and Data Center service management templates

This screenshot shows project templates in Jira Service Management Server and Data Center. In the "Service" project category, there's a choice between "basic," "IT service management," and "customer service" templates. Just like in Cloud, the templates contain different request types, workflows, integrations, and other settings.

Q Expert tip!

In a test environment, create a sample Jira project using each template in your application. Then you can try them all out, demo the differences for others, and experiment with configuration changes. Some templates contain a lot of additional configuration data. When it's time to create new projects in the production environment, be aware that you might need to do some clean-up or de-duping work.

Communication strategies for JSM implementation

Has this ever happened at your organization? The business team wants one thing, but the development team builds something else. Or the facilities manager asks for new office furniture, but the procurement manager orders the wrong size chairs.

Maybe the overall goal wasn't understood. Maybe it wasn't properly communicated. Possibly expectations didn't match reality. Or maybe it's all of the above.

Sometimes, the original idea isn't well defined, or individual requirements aren't clear to your team members. For example, to a software developer, JQL means "Java Query Language." To a Jira administrator however, it means "Jira Query Language." Add in language barriers, idioms, abbreviations, and unknown or hidden details and it's easy to see how communication can cause problems with implementation.

As an application administrator, I often feel like I'm the last to join planning and requirements discussions. Either everyone discusses the initiative and fails to communicate that information or, worse, they don't spend any time at all defining their goals and needs. This makes for wasted effort, dissatisfied stakeholders, and an undesired "solution."

Way too many times, I've said: "If I had known X, I would have done Y!"

To make sure I get the information I need to implement the right solution, I've created a little worksheet of example questions to ask stakeholders.

This helps them to determine, define, and communicate in a clear and actionable way. By the time they work out the initial details, I'm able to understand their needs and map them to implementation options in JSM.

Don't let unclear instructions or expectations derail your project. Use the example below as a guide to create a custom procedure for your organization. Use the table to collect, understand, and communicate needs so administrators build the right solution, team members can use it, and the desired outcome is accomplished.

Example scenario

A small health care company uses Jira to track software updates made to their medical devices. They collect customer feedback haphazardly, and don't manage it well. Feedback arrives by email, over the phone, via social media, and through other random channels. It's not stored in a central location or consistently acted upon.

The solution: The company wants to adopt JSM to collect and manage feedback from patients using their devices.

NOW WHAT?

Here's an example of the expectations I'd set and sample details I'd collect to implement a good solution.

For company leadership

EXPECTATIONS

As the owner of the company, you set high-level strategic priorities. You don't need to define all requirement details or implementation needs, but you do need to communicate the desired results.

QUESTIONS

Please define the major elements like:

- What is the overall goal?
- How does this initiative support the goal?
- How does the company define or determine success?
- What are the timeline, budget, resource, or effort parameters?
- And more

Next, it's time for team members to review the high-level information and determine how to accomplish it within the required parameters.

For support team leaders and members

EXPECTATIONS

It's important for support team leaders to understand how a process

works (or doesn't work) today vs. how it will work in the future. Think about what's changing, what's not changing, and how the changes will affect team members and goals. When processes change, you need additional training or documentation, too.

Support team members, on the other hand, need to understand how any changes impact daily work and responsibilities. For example, how will you receive, share, and manage the feedback? What are you expected to do with this information? How will you meet organizational expectations as well as customer expectations?

QUESTIONS

Managers and team members should work with leadership, business analysts, and other stakeholders to answer questions like:

- What methods can customers use to submit feedback? List any channels to use or avoid.
- What happens to feedback once it arrives? Determine who needs to act, what they need to do, and how often. Does the information need to be shared with others?
- How do you measure customer sentiment? Define characteristics of a satisfied customer vs. a dissatisfied customer.
- How will you measure the success of this initiative? Provide specific metrics.

Use this information to draft a set of requirements for the JSM administrator.

For application administrators

EXPECTATIONS

The application administrator must then review the requirements and translate them into JSM implementation instructions.

It's important to figure out what the application can do, what it can't do, and what solutions are available (if any) to bridge those gaps.

QUESTIONS

How will each requirement be accommodated?

- Are apps or integrations needed to extend native functions? Include some options from the Atlassian Marketplace.
- Will the implementation plan meet the parameters or constraints? In other words, can the project be completed on time and within budget?
- What's the near-term and long-term impact to application users? To application maintenance?
- How can the implementation be tested? Determine specific test cases and enlist stakeholders to participate in the testing process.

Translate requirements to JSM instructions

Here's how I would translate JSM requirements based on the questions above and some sample answers.

Question	Answer	JSM Implementation
What is the overall goal?	To measure customer satisfaction and improve our products	 Create and configure a "Jira Service Management" support project
How does this initiative support the goal?	By collecting feedback, we can determine whether the customer is satisfied, deliver new features, and fix defects.	 Create issue types and request forms for feedback collection, feature requests, and bugs.

Question	Answer	JSM Implementation
How is success defined or determined?	50% of customers submit within 5 days of receiving their device. 75% of feedback collected is positive.	 Create a custom field to collect the delivery date and add the field to the portal request form. Create a JQL query to count the total number of feedback submissions. Compare that number to stats in the sales application. Consider creating a filter subscription or automation to send the most recent count information to stakeholders. Create a scoring system to distinguish positive and negative feedback. Options: Create custom structured fields to ask specific questions and collect "scores." A single select field, with preset selections, will provide easier to calculate results than a free text field. Leverage JSM's built-in "Customer Satisfaction" email survey. Install a survey app from the Atlassian Marketplace Create an automation to send customers a survey managed outside of JSM. e.g., SurveyMonkey, Constant Contact, etc. Use a free text field to collect feedback. Have a representative from the support team manually classify it as positive or negative. Which option is preferred?
What are the timeline, budget, resource, or effort parameters?	Needs to be implemented by the end of Q2. Budget: \$1,000	Approx. 10 man hours for JSM implementation + \$25/month for survey app (based on current user count) + 3 man hours to create and deliver training sessions (plus one hour for each service team member to attend training) = Estimated total: \$970

Question	Answer	JSM Implementation
What methods can customers use to submit feedback? List any channels to use or avoid.	Submit from the company website and the JSM customer portal. In the future, we may add support via email too.	 Configure the JSM project's customer portal settings Enable the widget feature and embed it on the company website Disable the "email requests" ability
What happens to feedback once it arrives? Determine who needs to act, what they need to do, and how often. Does the information need to be shared with others?	All submissions should receive a response within 24 hours. Support team members should use standard pre-written copy to thank customers for their submission. For feature requests and bugs, support team members should create a new Jira issue for the development team, and link it to the support request.	 Create a "time to first response" service level agreement (SLA) and set the goal for 24h. Create a queue to segment issues about to miss the response deadline. Use a query like: "Time to resolution" < remaining ("2h") Consider an automation to alert the assignee or support team lead of requests that are about to miss the response deadline. Add a response template app from the Atlassian Marketplace (Approx. an additional \$37.50 per month.) Build an automation with a manual trigger that creates a new issue in the development project, automatically links the issue to the request, and includes specific information from the original request.
How do you measure customer sentiment? Define characteristics of a satisfied customer vs. a dissatisfied customer.	Use a simple 1-10 scale, with 1 meaning "very dissatisfied" and 10 meaning "very satisfied." Ask the user about the speed of delivery, ease of setup, and overall satisfaction.	 Create three custom fields to collect the speed of delivery, ease of setup, and overall satisfaction responses. Use "single select" fields with selection values: 1, 2, 310. Should any or all fields be required? Also suggest using a free text box to collect any additional information the customer would like to provide.

Question	Answer	JSM Implementation
How will you measure the success of this initiative? Provide specific metrics.	Customers are able to submit feedback, feature requests, or bug reports by the end of Q2. All apps were setup in time for launch. The project was completed within budget. Support team members were trained on the new process.	• n/a
Are apps or integrations needed to extend native functions? Include some options from the Atlassian Marketplace.	n/a	 For survey functionality, suggest using <u>Surveys for Jira</u> or the built-in "Customer Satisfaction" function For standard communication, suggest using <u>Canned Responses Pro Templates</u> for Jira
Will the implementation plan meet the parameters or constraints? In other words, can the project be completed on time and within budget?	n/a	 Yes. The only data JSM doesn't have is access to stats in the sales application. It is possible to use an app to access that external database and use an automation to calculate the feedback percentage received. Taking this further, an automation can alert stakeholders via email or chat message when the percentage is less than the goal.
What is the near-term and long-term impact to application users? To application maintenance?	n/a	 Create service accounts for a test customer user and a test support team user Encourage stakeholders to create test feedback, feature requests, and bugs Verify all automations execute correctly and as expected Verify the "time to first response" SLA is 24h and functions as desired

After the implementation options and challenges are identified, the application administrator should collect answers for any remaining questions and discuss plans with all stakeholders. Don't be surprised if implementation notes create other requirements or considerations. Take the time needed to address these items. **Don't start the implementation until all questions are answered and the plan is finalized**. A solution is always better when you consider the details up front and all elements are clear to everyone involved.



CHECKLIST

Rachel's 10 Tips for Managing Support in Jira Service Management.

Use these tips and recommendations to implement and get the most benefit from JSM.

1. IMPROVE YOUR PROCESS

Jira Service Management is just a tool, and ITSM is just a way to manage service. No tool, methodology, or framework can make up for bad process and leadership. If your organization's process is a mess outside of JSM, that mess is likely to creep into JSM as well. Before implementing new software, try to optimize your process, workflow, and communication.

2. TRAIN USERS AND SET EXPECTATIONS

It's important to set clear customer support expectations up front. For example:

- How should customers request support? JSM can accept requests in multiple ways. (i.e., via the customer portal, email, an embeddable widget, an app or integration, etc.)
- How will you communicate how long certain requests might take to fulfill? Will you use on-screen messaging, automatically update users via email or chat, or use service level agreements (SLAs) to display a time goal?

You'll need to train your users how best to work with your team and your software.

3. ADAPT TO THE CUSTOMER

Start with just a few JSM request types. Don't add more until there's a real business or reporting need. And always customize form field and status names with language the customer understands and expects.

- For example, in the customer portal, change the label for the default "Summary" ("request title") field to "How can we help?" or something else that's customer friendly and instructive.
- Where necessary, provide a field description or a formatting example.
- Don't make the customer guess what information you need or how to provide it.

4. MEASURE SUCCESS

Use JSM to measure how well you're meeting deadlines, goals, and performance metrics. But make sure the support team focuses on the quality of support, not just the quantity of requests completed.

5. AVOID OVER-CUSTOMIZATION

Every time you create a new service desk, change a configuration setting, or install an app, consider the health of the application as a whole. Decisions you make today will impact maintenance, performance, and usability in the future.

6. USE QUEUES TO STAY ORGANIZED

Use JSM queues (lists of related issues) to organize and filter requests. When creating these shared views, consider how the team delegates and prioritizes work. For example:

- If work is divided by area of the business, create a queue for each area.
- If work is ranked by priority, create a queue for the most important requests.
- If your organization has an important initiative going on, create a queue for requests relating to that, too.

Make it easy for the support team to respond to the right requests at the right time.

7. PERIODICALLY REVIEW PERMISSION SETTINGS

ISM has multiple levels of permissions for the entire application, projects, issues, requests, workflows, and apps. This flexibility is fantastic, but it also makes it easy for admins to give the wrong access to the wrong users. Create a test account for each new type of user, group, or role. Verify the capabilities and permissions for customers, agents, participants, and administrators work as expected. Audit all application permissions on a regular schedule. For example, check all permissions each quarter.

8. INTEGRATE APPLICATIONS

Get the most value out of your Atlassian products by connecting them. Help your teams communicate by linking JSM requests to their related Jira Software issues, Jira Work Management plans, Insight assets, Opsgenie alerts, and Confluence pages. Use WebHooks, the REST API, or apps to integrate internal applications and non-Atlassian software, too!

9. INVITE OTHER TEAMS

Jira Service Management isn't just for IT teams. JSM can help any team with customers to support. For example, the HR team's customers are the employees, and employees have plenty of requests to manage! This team — and any team with a support function — can benefit from using Jira Service Management.

10. EXTEND WITH APPS

Make JSM even more powerful by adding apps that extend its capabilities. For example, add the <u>Canned Responses Pro Templates for Jira app</u> to quickly and consistently respond to similar requests. Use the Chat for Jira Service Management app to collect real-time details, deliver a fast resolution, and improve customer satisfaction. Use automation to remove manual steps and keep work moving forward. Continuously improve service quality with <u>Time</u> to SLA. By utilizing different reports and metrics, teams can analyze their progress and get better over time.



Resources

Related apps to help you boost your service desk.

- Collect customizable and brandable surveys with <u>Surveys for Jira</u>.
- Craft reusable support wording with <u>Canned Responses Pro Templates for Jira</u>.
- Use the Polls for Confluence app to create custom polls and surveys.
- Use <u>Easy Forms for Confluence</u> or <u>Survey and Vote Macros</u> to bring feedback capabilities to Confluence Data Center.
- Collect customizable and brandable surveys in Jira with Surveys for Jira.
- Use the Reports and Timesheets for Jira app to simplify reporting.
- Get data-driven insights and visibility with the **Power Bl Jira Connector** app.
- Install the Enhancer Plugin for Jira app to boost reporting capabilities (and much more).
- Use the <u>Chat for Jira Service Management</u> app for fast resolution and high customer satisfaction.
- Add service desk customers and organizations in bulk with the <u>Jira Command Line</u> <u>Interface (CLI)</u> app.
- Push customer information from Salesforce to Jira with the <u>Power Salesforce Connector</u> for <u>Jira</u> app.
- Add workflow behaviors and conditional logic with these apps:
 - ° Jira Miscellaneous Workflow Extensions (JMWE).
 - Power Scripts Jira Workflow Automation.
 - JSU Automation Suite for Jira Workflows.
 - Power Utilities for Iira Workflow toolbox for admins.
- Install the Electronic Signatures app and never sign a paper document again.
- Use the <u>Time to SLA</u> app for additional search and reporting features.
- Read How to optimize your IT Service Management with Appfire apps to enhance communication, time tracking, and bring additional service management capabilities to your application.



About Appfire:

Appfire is a global authority in the Atlassian ecosystem. Appfire's popular solutions help teams with Workflow Automation, Product Portfolio Management, IT Service Management, Document Management, Business Intelligence and Reporting, Administrative Tools, Agile, Developer Tools, and Publishing. The company has the most widely adopted portfolio of apps on the Atlassian Marketplace, with 225,000 active installations worldwide. Learn more at www.appfire.com.

Notice of Liability

The information in this book is distributed on an "As Is" basis, without warranty. The publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained in this book. Use of the information and instructions contained in this work is at your own risk.

© Copyright 2022 Appfire Technologies, LLC. All rights reserved.