Solution Delivery Blueprint

Enterprise Service Management for Software teams

(Jira Software + Jira Service Management)

MARCH 2023

Table of Contents

About this guide

The guide provides a comprehensive overview of how to deliver and implement Atlassian tools and IT Service Management (ITSM) practices using a standardized delivery process while getting an opinionated view and leveraging best practices. It is written for those of you that are about to set up a Jira Service Management (JSM) implementation project.

The target with this guide is to make ITSM implementations easier, with less pain and getting happy customers. The guide will be regularly updated with new information and feedback to ensure that it stays current and effective.
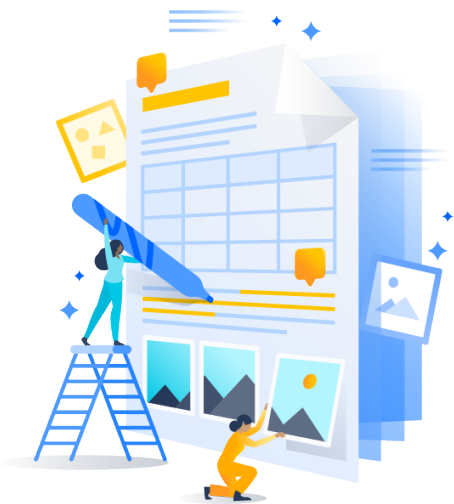
# Use case: Enterprise Service Management for Software Teams

Software teams working with, and maintaining, products and services.

Solution description

You will need to understand the process you are about to implement before you start making any changes. Keep as much as possible of the out of the box functionality to get started, and understand the team which is going to use the solution and what they care about.

Jira Service Management for software teams provides the best functions that software development teams need to stay on top of incoming issues, intake of feature requests, bug activities while working with the tools that they know.

Many software teams prefer to work with chat tools like Slack or Teams, rather than email, and JSM has integrations for that purpose. This allows your organization to reach the software team directly from chat, the portal or email, while the software team can respond directly from the chat.
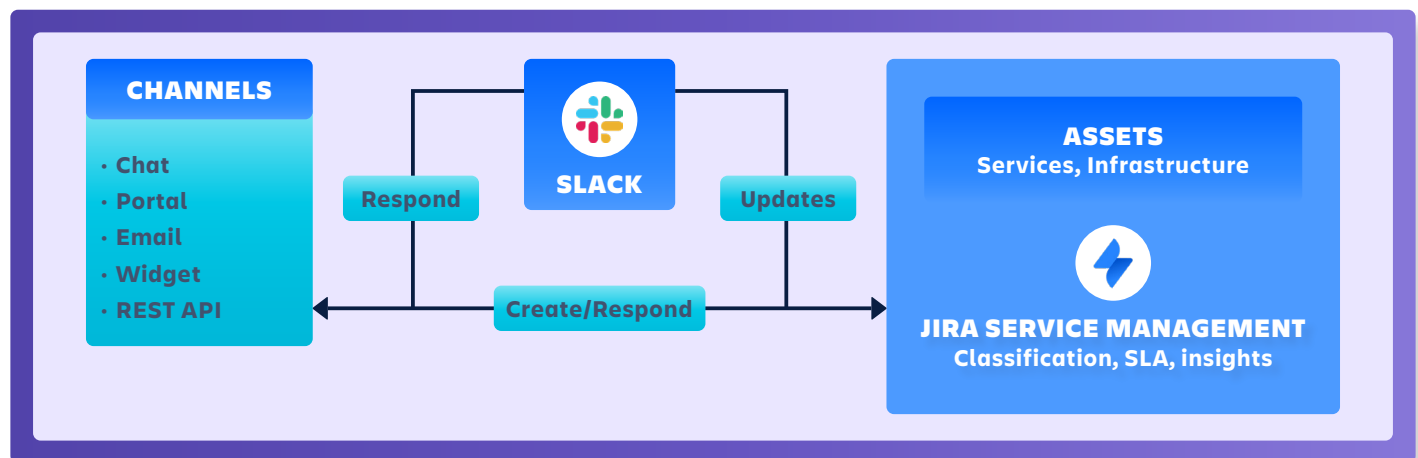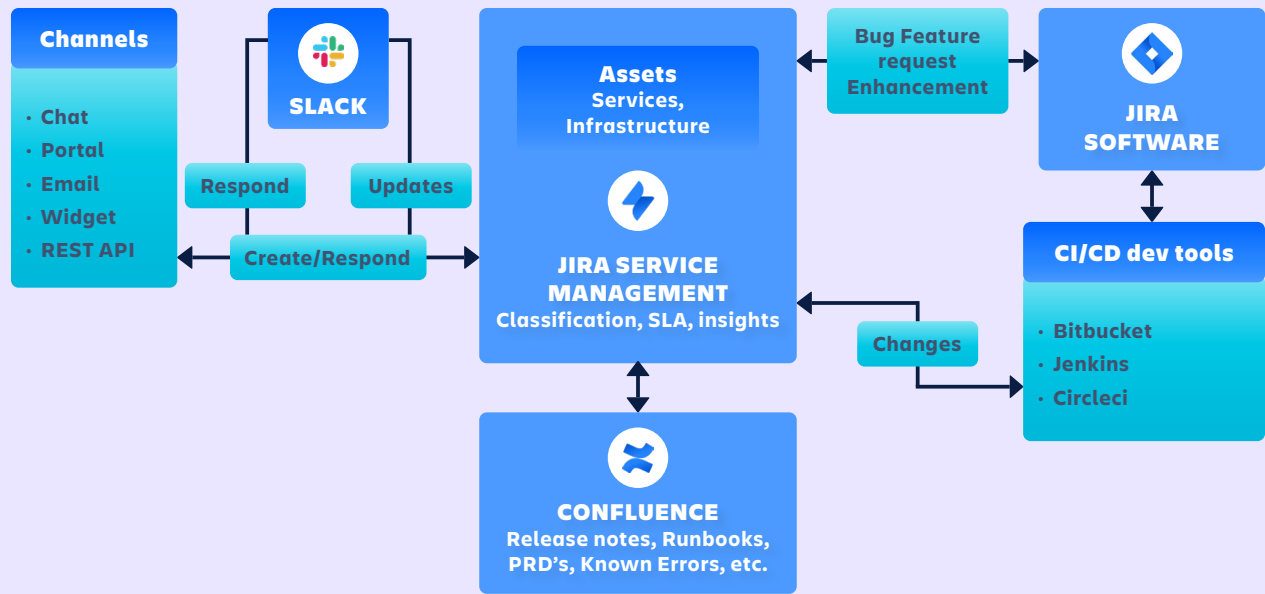

Software teams also want to improve their products and services and need feedback from their users. On top of having available request types in the portal, such as "feature request", "enhancement" or "report a bug", It's also useful to use **widgets** or **issue collectors** to get that additional feedback from users who don't normally go to Jira Service Management.

Software teams want to focus on writing code and automate change requests as much as possible. Being able to deploy code without manually going to Jira, creates a much better user experience and allows the teams to manage their time more efficiently.

This also gives your operations a complete audit trail of the changes deployed and is critical for compliance and risk handling.

Finally, software teams love to collaborate with Confluence and use it to document their Release notes, **Product Requirement Documentation** (PRD), Known Errors and FAQ's.

SOFTWARE TEAMS PREFERRED TOOLS AND WAYS OF WORKING

Channels
- Chat
- Portal
- Email
- Widget
- REST API

SLACK

Respond / Updates / Create/Respond

Assets
Services, Infrastructure

JIRA SERVICE MANAGEMENT
Classification, SLA, insights

Bug Feature request Enhancement

JIRA SOFTWARE

CI/CD dev tools
- Bitbucket
- Jenkins
- Circleci

Changes

**CONFLUENCE**
Release notes, Runbooks, PRD's, Known Errors, etc.

Added value

There's always a need to focus on the value that triggers the interest of the stakeholder you are talking to, and for that you need to understand their roles and their needs, whether they are incident managers, product owners, management or developers in a team.

1 JSM for software teams brings visibility to the whole organization, from feature requests, bugs or support tickets down to the code. Having everything in one place (single pane of glass) also helps with prioritization and planning.This is something that management and product owners really value and makes it a good selling point for Partners.

2 JSM for software teams allows teams to collaborate and automate redundant tasks, which saves time and allows the teams to work on the things that matter.This is something the teams really value and makes it a good selling point for Partners.

3 JSM for software team leads to faster Mean Time To Resolve (MTTR) by increasing visibility on what has been recently deployed, in the new **investigation view**:
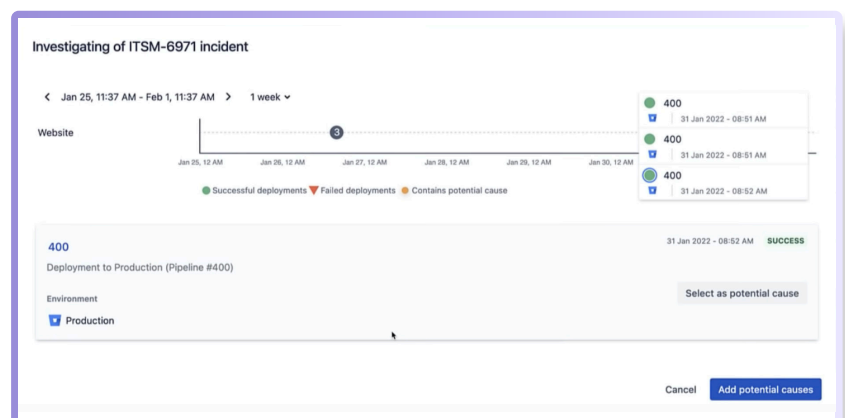
4 JSM for software teams also helps with preventive actions, when needed, by using the built in **Post Incident Review**. Incident Managers love to hear this and this is a great selling point for **Partners**.

5 Finally JSM comes with a reports engine that allows software teams to analyze and improve. For example, how many bugs have been reported for a specific version.
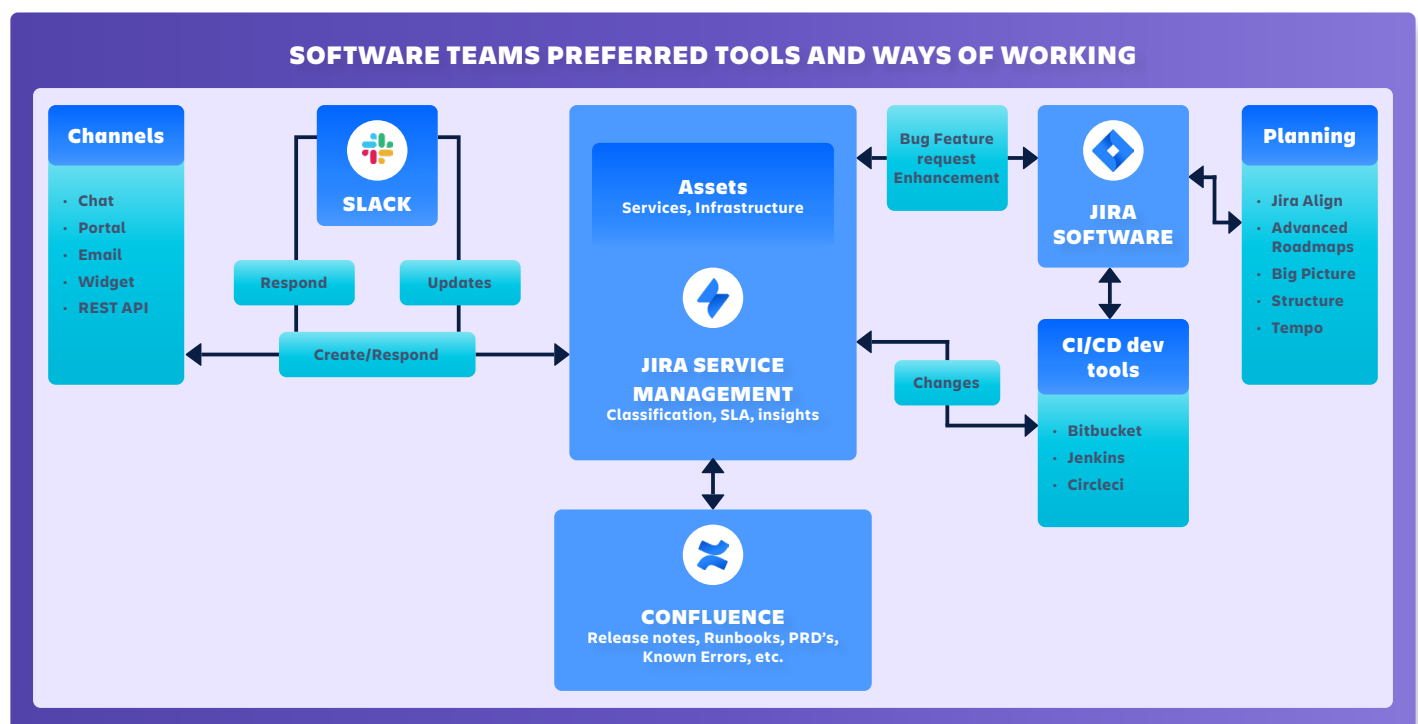
Solution overview

For every use case, there are three implementation steps.

1		**Knowing**	discovering the existing practices that the team uses.
2		**Understanding**	diving deep into the existing practices and identifying areas of improvements.
3		**Creating value**	improving the existing practices.

Solution workflow

In real life, software teams working with both Jira Software (JSW) and JSM, will be part of an organization that wants to track and plan the progress of multiple teams and projects, which is when you start involving products such as Jira Align and Advanced Roadmaps.

The diagram below shows the flow of how the products are used from a software team perspective.

SOFTWARE TEAMS PREFERRED TOOLS AND WAYS OF WORKING

Channels
- Chat
- Portal
- Email
- Widget
- REST API

SLACK
- Respond
- Updates
- Create/Respond

Assets
Services, Infrastructure

JIRA SERVICE MANAGEMENT
Classification, SLA, insights

Bug Feature request Enhancement

JIRA SOFTWARE

Planning
- Jira Align
- Advanced Roadmaps
- Big Picture
- Structure
- Tempo

Changes

CI/CD dev tools
- Bitbucket
- Jenkins
- Circleci

CONFLUENCE
Release notes, Runbooks, PRD's, Known Errors, etc.

- ✓ End users register tickets in their preferred channel. Software teams respond in Slack which also updates JSM.
- ✓ Bugs, enhancements and feature requests are linked to Jira Software which in turn are tracked in Jira Align or Advanced Roadmaps.
- ✓ Developers deploy code which generates automated changes in JSM.
- ✓ Release notes, runbooks, PRD's, known errors and more are documented in Confluence.

Admin configuration step by step

There's always a need to focus on the value that triggers the interest of the stakeholder you are talking to, and for that you need to understand their roles and their needs, whether they are incident managers, product owners, management or developers in a team.

1 Start by creating a new project, using the **IT Service Management template**.

2 Go to your newly created project and click on **Services** to set up the right products or services that the software teams are supporting. Make sure to add the right Tier (Service Tier), Change approvers (Groups) and Repository to your service.

3 Set up **deployment tracking** with the right CI/CD tool:
- ✓ Select the correct environment/s and request type
- ✓ You can also **enable deployment gating** if the customer needs to allow or prevent deployments at specific points in the change management process.

4 **Integrate Jira Cloud and Slack**.

5 **Setup SLA** for your project.

6 **Setup Queues** for your project.

7 Modify the predefined **screens and fields** to match the need.

8 Add and modify **Forms** that can be used for the different workflows.

9 Document each process with the work routines, so that the agents can fulfill the requests and follow the runbooks in Confluence.

10 **Modify the workflows** and choose what statuses are available for the **request types**.
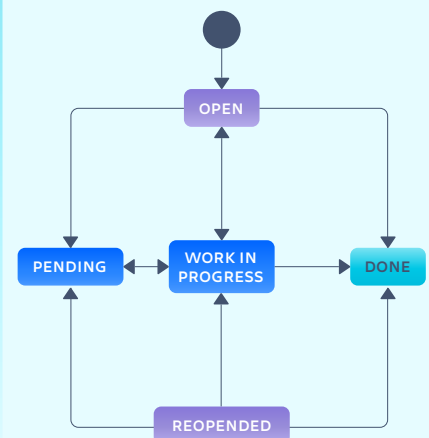
Below is a typical workflow that comes with the ITSM Template for Change Enablement:
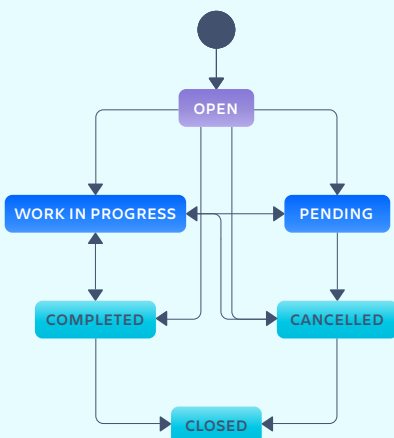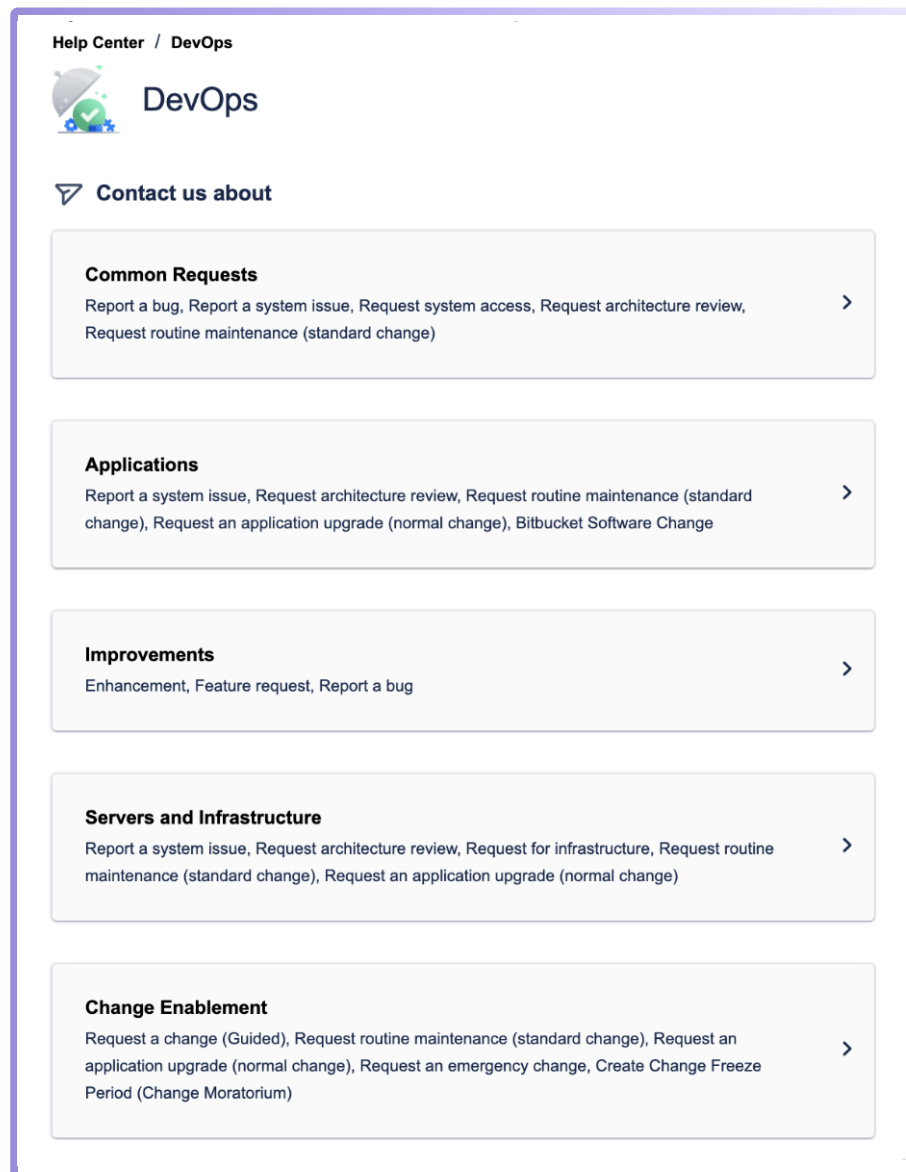
Below is a typical workflow for Incidents:
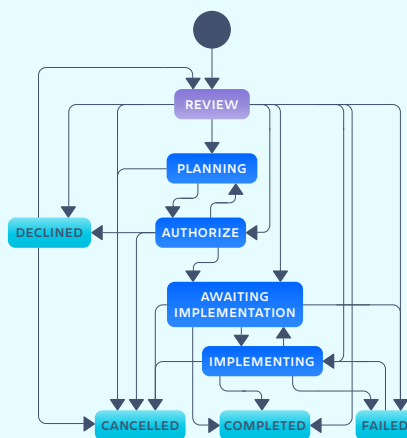
Below is an example workflow for a bug, enhancement or feature request. The Pending state is often used while awaiting for a new release:

11 Configure the portal, make sure to group the request types into logical categories for your end users. Here is an example of how it can look like:

12 Setup **widgets** or **issue collectors** to collect feedback.

13 The template has some **automations** that are handy, however make sure to review them and adapt them to your needs. See below for some examples.

Automation name	Comment
When a change management request is created then update change type and risk	Update according to your company's risk matrix, and update any standard/normal changes to fit the need.
Add form or runbook to your workflow	Make sure to add helpful forms or runbooks for the different processes, there are hundreds of templates to get started.

Automation name	Comment
Approved Normal Change raises JSW issue	Here's an example of a change that creates a ticket in a JSW project after it has been approved. When: Issue transitioned FROM Authorize TO Awaiting implementation Request Type equals Request an application upgrade (normal change) Then: Create a new Story in Platform Development And: Link issue to Most recently created issue
Major Incident resolution raises linked Post Incident Review (PIR)	An automation that checks for major incidents and creates PIR's. ITO: Major Incident resolution raises linked PIR Rule details Audit log When: Issue transitioned TO Resolved, Completed Issue Type equals [System] Incident If: matches {{issue.Major incident}} equals MAJOR_INCIDENT Then: Create a request Post Incident Review in IT Operations And: Link issue to Most recently created issue
Inform users of fixes in new releases	An automation that checks for Jira Software issues that are closed and comments on any linked JSM ticket with information about the fix. For Linked issues Types: relates to *Hi {{issue.fields.reporter.displayName. split(" ").first}},* *We just wanted to let you know that we've just released a new version which resolves {{triggerIssue.key}} ({{triggerIssue.summary}})!* *Cheers,* *Automation for Jira*

14 Finally you can also consider setting up a **mailbox** for incoming tickets.

You are now all set with most of the needed functions, congratulations!

Additional tips and tricks

To go beyond what is available out of the box and expand the functionality we recommend looking at the powerful marketplace for apps. One example can be modifying the look and feel of the portal, for that we often use apps like Refined and Theme Extension for Jira Service Management.

Software teams can benefit from adding release notes, identified bugs for their services and feature requests currently being considered, as well as additional links and an improved portal layout.

Another example, expanding beyond the out of the box functionality, is being able to hide specific request types that should not be visible to everyone.