

Improve Your ITSM Approach Using Jira Service Management (JSM)

Jira Service Management (JSM) began when Atlassian research discovered that many of its customers were using apps in Jira Marketplace to create ad hoc service management desks for their internal users. Atlassian integrated, normalized, and added capabilities to these applications to produce JSM. JSM enables customers to customize Jira's core platform to accommodate a variety of workflows. Through JSM, customers can configure and integrate Jira frictionlessly within their work environments.

Shortly after the introduction of JSM, [Pink Elephant confirmed that Jira Service Desk](#) supports the four main pillars of an Information Technology Infrastructure Library (ITIL): incident, problem, request, and change management. ITIL is the standard underlying Information Technology Service Management (ITSM). JSM is ITIL-certified and provides customers with an integrated package that supports service management.

JSM enables you to understand the activity involved in the operations and maintenance of your applications. While essential, these activities take time away from developing the applications that add value to your business. JSM provides developers with a single place to find and act on information, which reduces the time from detection to remediation.

How JSM Improves on ITIL

JSM does not replace ITIL. It allows you to automate and administer best practices in developing software. Software is seldom perfect when it first leaves the developers' hands, so JSM enables you to remediate application issues by organizing and delivering information aiding the solution when needed.

JSM provides methods to improve the velocity of the various work streams of ITSM. Let's take a look at some of these different areas and explore how JSM improves ITSM approaches.

Request Management

With JSM, a team can build a service desk customized to provide the user (either a customer or employee) with the information they need to start a request and have their requisition information routed to the correct group to respond.

Furthermore, JSM [form builder](#) allows the team to build forms that collect information they need to complete the request. The advantage is that this process eliminates the need to search for information or call the customer. It also sets up

a communication channel between the servicer and the customer to advise them of the request's status or obtain other information.

Request management customizes, integrates, and automates the request processing workflow. Once a team has completed a task, the workflow passes the request to the next team that needs to act on it. JSM request management can provide the customer with self-service forms and processes that do not require human intervention. Check out [this article](#) for a deeper look into the features of JSM Request Management.

Incident Management

Incident Management is another ITSM workstream that can benefit from JSM. Incident responses need to be customized to fit the organization. JSM's automated filtering of event and response triggering allows incidents (and information surrounding it) to be routed to the responsible team. Then, customized on-call schedules alert the correct set of individuals.

Escalation patterns ensure that alerts can be handled quickly by incorporating the source and urgency into the alerting process. This process can alert stakeholders, such as account management, to incidents that could affect customer relationships or SLAs. In addition, major incidents, as defined by the organization, can be automatically escalated.

Problem Management

Where incident management is about fixing an immediate issue, problem management is about diagnosing and preventing incidents from reoccurring, thus strengthening the systems. Problem management starts with identifying related incidents. From there, JSM collects tickets from the related incidents and their retrospective reports and reviews the development team backlogs to determine whether the problem is on their radar. This avoids duplicate workstreams. Furthermore, JSM can publish a report to Confluence to let others know the problem is under investigation or to let them know of possible solutions or workarounds.

JSM problem management also sets about identifying potential root causes of the problem and investigating likely causes, such as a service outage or underlying software issues. The appropriate developers are alerted if the problem is due to code commits. In short, JSM problem management looks at all the potential causes of the problem and brings together the people and information it takes to resolve it.

Change Management

Change management is another area where JSM can improve your processes. All change requests carry some risk. In many cases, the risk is minimal due to the number and criticality of the systems involved. In other cases, the risk can be higher, especially if the change affects multiple applications or many changes occur simultaneously. JSM assesses the risk by looking at various factors, including systems involved, the change calendar, and lists of affected services. This information generates a list of approvers for the change.

In addition, JSM monitors Bitbucket pipelines and other development tools to automatically evaluate the change without having a developer create a manual change request. This monitoring lets developers focus on development while the request is implemented or submitted for further approval. JSM provides tools to evaluate risk and minimize the amount of human intervention, increasing the velocity of software deployment.

Asset Management

Managing assets can be a chore for IT operations. Knowing an asset's location in the age of mobile assets can be particularly tricky when people relocate. There is also the question of which assets need replacement when they've reached the end of their useful lives. Which software is properly licensed and updated? Asset management has a great deal of complexity, but that's where JSM comes in.

JSM can search your network for assets and discover those which are not in your inventory. You can reassign assets to avoid unnecessary, additional purchases with a complete list of assets, properties, and ownership. JSM asset management can be set up for personnel to self-serve by requesting assets and routing the requests to the appropriate approvers, speeding up the process and reducing personnel costs.

Configuration Management

Configuration management is an important but potentially time-consuming operation. The IT infrastructure is constantly changing with software updates, security patches, and equipment replacements. Software is interconnected so that changes to one part may affect many others. To understand and minimize the risk involved in configuration changes, DevOps needs to be aware of these factors.

JSM can scan the network and report each node's configuration, dependencies, and status. JSM can also create and track issues when unauthorized changes are discovered. It improves ITSM by monitoring, analyzing, and reporting on your infrastructure configuration.

Knowledge Management

Knowledge management is another key part of ITSM that JSM handles. A key to knowledge management is the ability for the customer to self-serve, avoiding a call to a help desk. Based on the questions that reach the help desk, JSM can identify gaps in content and people's ability to locate it, improving the self-service function over time. JSM also uses machine learning to improve search results, providing users with the most relevant articles based on prior click-throughs.

In addition, JSM's connectivity to Confluence provides users with formatting and editing capabilities enabling them to document new knowledge without leaving JSM. This integration makes it more likely that key information will be documented immediately rather than put off and possibly lost.

Check out [this article](#) for more information on JSM features.

Integrations

JSM provides out-of-the-box ITSM processes that support Jira software and a self-service interface and automation of ITSM processes. In addition, JSM can link to service tickets in Jira to guide the prioritization of development requests. The development teams can access all the information and comments on the ticket. In addition, they can add their comments and questions to the same ticket so that teams across the organization can collaborate. For a detailed discussion on the Jira/JSM integrations, see [this article](#).

For users of Slack and Teams, JSM offers conversation-based ticketing. The integration allows users to open a ticket from [within Slack](#) or [Teams](#). Once open, all the participants can access the chat and comments made. JSM users can use Teams or Slack from within JSM to have conversations with each other or the customer without leaving JSM.

Conclusion

JSM provides out-of-the-box support for ITSM processes and abilities like self-service equipment provisioning, automated approval, and change and configuration management. All the features discussed here remove burdens from your DevOps teams so they can focus on value delivery.

If this sounds like a potential fit for your needs, check out [Methoda](#) to learn more about JSM and how to configure it to best serve your organization.