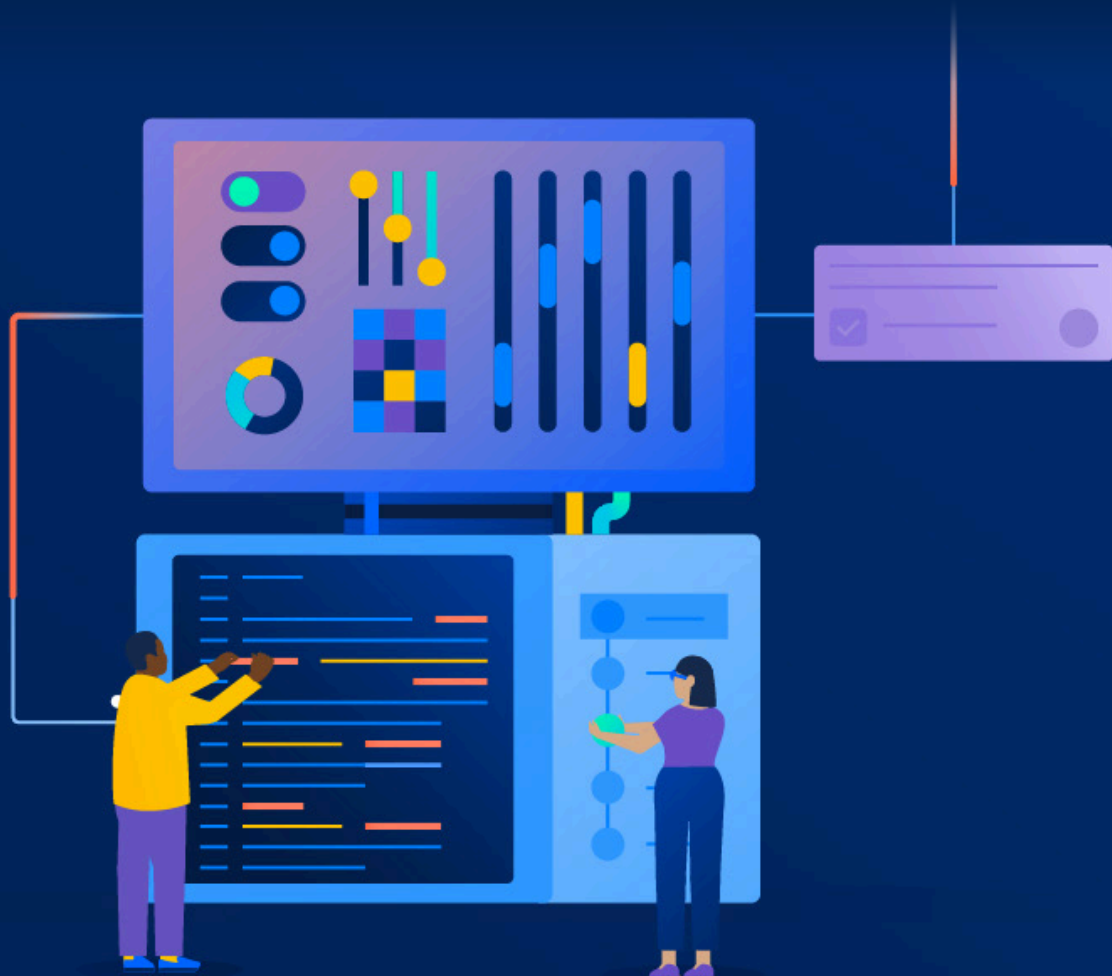


# Solving The Pain Points Of Traditional ITSM Upgrades

A Product Owner learns firsthand about  
alternatives to a traditional ITSM  
system upgrade

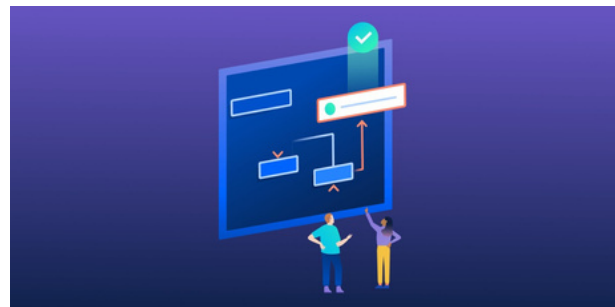


# Table of Contents

<b>In brief</b>	<b>2</b>
<b>The pains of upgrading a traditional ITSM system</b>	<b>4</b>
<b>Seeking relief – a better approach to ITSM systems</b>	<b>5</b>
<b>A healthier approach to ITSM system upgrades</b>	<b>7</b>

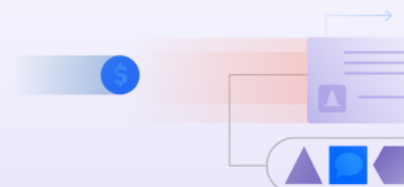
## In brief

In this feature you'll learn of the challenges faced by a Product Owner in upgrading a traditional IT Service Management (ITSM) system, and their subsequent exploration of Atlassian's Jira Service Management as a more flexible solution for ITSM system upgrades.



### **Read On, To Discover:**

- 1** How the Product Owner encountered significant challenges when upgrading their traditional ITSM system. Learn why it was a complex and time-consuming process, with frequent disruptions, and perceived low value to end users.
- 2** Learn why one of the main benefits of Atlassian's approach to Jira Service Management upgrades is that it follows an Agile methodology, allowing for more flexibility, collaboration, and efficiency compared to legacy ITSM vendors' waterfall approach.
- 3** Hear why many organizations are experiencing similar pains with ITSM system upgrades, with some opting to "reboot" legacy systems to mitigate the issues.
- 4** Find out how Jira Service Management offers continuous improvement through frequent feature releases, a loosely-coupled architecture for efficient code delivery/upgrades, and a true cloud product that scales easily and provides enhanced security and compliance. This approach aligns with the need for a more agile and non-disruptive pace in digital transformation efforts.
- 5** Discover why the Product Owner turned to Atlassian's Jira Service Management. They were initially sceptical, but found it to be a more flexible and efficient solution for ITSM system upgrades, with features such as easy configuration and out-of-the-box ITIL practices.



## A Product Owner learns firsthand about the alternatives available to a traditional ITSM system upgrade

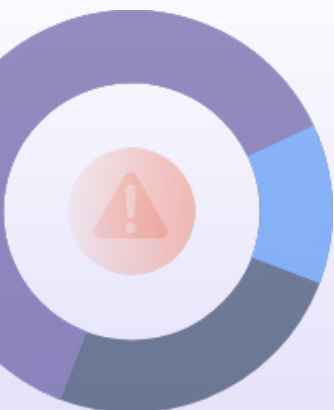
I was a Product Owner for a Scrum team that developed capabilities for several service management practices using a legacy ITSM system. As the last sprint of the calendar year wrapped up, my team enjoyed an innovation sprint in which we could learn and experiment with new functionality. I used this

innovation sprint to look at our ITSM system's new release. At first glance, there appeared to be tons of new functionality. On closer reading, the new functionality wasn't useful to our stakeholders and would result in more technical debt.

“I poked at the Assets feature in Jira Service Management for a while, and it seemed easy to configure and load data. Then I started creating projects and service assignment groups. After only a few hours of configurations (and glancing at any documentation), I had developed a working service catalog and service desk that included IT assets and CIs and major incident capabilities.

My head started pounding in anticipation of the numerous meetings with our process owners and other stakeholders. Just the thought of explaining the content of the new release, the disruption to our development schedule, and what would ultimately be only

a small useful impact of all this new work stressed me out. Regardless of how many features we could consume, the tedious process and lost productivity of upgrade season were always the same.



# The pains of upgrading a traditional ITSM system

In this feature you'll learn of the challenges faced by a Product Owner in upgrading a traditional IT Service Management (ITSM) system, and their subsequent exploration of Atlassian's Jira Service Management as a more flexible solution for ITSM system upgrades.



## At A Glance:

- 1** Upgrading software can be a daunting process, as it involves numerous complex steps that lead to disruptions and frustrations for end users.
- 2** The challenges of software upgrades are not isolated, with many organizations experiencing lengthy, disruptive upgrades that impact productivity.
- 3** Some organizations are even considering “rebooting” their legacy systems to mitigate the difficulties associated with software upgrades.

For those who haven't dealt with this, some of what I describe may seem exaggerated, so below is a snapshot of my team's upgrade process. This involved many tasks that I've distilled into eleven broad steps across three phases from pre-release to release to post-release.

<b>Pre-Release</b>	<ul style="list-style-type: none"><li>• Product owner reviews release notes, inventories new features and fixes</li><li>• Dev team installs release in sandbox to use and document features</li><li>• Product owner meet with process teams to agree on useful features</li></ul>
<b>Release Process</b>	<ul style="list-style-type: none"><li>• Ops team clones all environments in preparation for upgrade ( code freeze for any new dev work)</li><li>• Dev team to package release code. Installs the release in Dev environemnt, deactivates conflicting code, and performs preliminary code tests. Then migrates release to QA</li><li>• QA team tests functionality using a testing framework; this testing addresses specific functionality, but not end-to-end process testing. after QA testing is successfully completed, Dev migrates the code to UAT</li><li>• Product owner and process teams test end-to-end processes in UAT environment. after UAT testing is successfully completed, Dev team documents the release procedures for products.</li><li>• Dev coordinates with other scrum teams to install code in Pre-Prod environment and make any needd updates; QA team tests Pre-Prod environment</li><li>• Ops team installs code in Production; QA team and process teams test Prod</li></ul>
<b>Post-Release</b>	<ul style="list-style-type: none"><li>• Dev team fixes any newly discovered post-production bugs</li><li>• Dev team installs tests, and deploys and required hotfixes</li></ul>

Just looking at this makes me tired. And it doesn't even capture the whole picture. Teams managing upgrades spend significant time and effort addressing organizational change management concerns. End users become frustrated with the twice-yearly disruptions to their service deployments which they perceive to be of little value.

I used to think that my team's issues with upgrades were isolated, but after attending group meetings, and reading blogs, I realized user that a painful upgrade process was ubiquitous.

Some customers complained of upgrades taking over eight weeks and grumbled about lost productivity for their teams.

In some cases organizations are going so far as to "reboot" their legacy ITSM systems to ease upgrades. For example, customers are still trying to figure out how to consume ServiceNow's Next Experience UI that was included in the San Diego release March 2022. Some enterprise customers are opting to switch off the functionality until they can determine the impact to their users.

## Seeking relief – a better approach to ITSM systems



### At A Glance:

- 1 When faced with a choice between enduring ITSM challenges or seeking a more robust solution, Atlassian's Jira Service Management is worth looking into.
- 2 What is perhaps surprising about Jira Service Management is that it is easy to configure, and quick to create a functional service catalog and service desk quickly.
- 3 Atlassian's approach to Jira Service Management upgrades is Agile, providing flexibility and ongoing feature releases, while legacy ITSM vendors often use a Waterfall approach with required yearly updates and tightly-coupled architectures.

The way I saw it, I had two options. I could either accept the inherent challenges of our ITSM system and fill my desk drawer with antacid and ibuprofen tablets, or look for a more robust, flexible solution. In frustration I took to Google, read analyst reports, and consulted with friends in the ITSM space, which led me to Atlassian's ITSM solution, Jira Service Management.

Frankly, I was skeptical that it could alleviate our issues. I have worked with enterprise ITSM systems for more years than I care to count, and I have also worked in Jira Software for a few years. So I felt confident Jira Service Management wouldn't address our issues.

I poked at the Assets feature in Jira Service Management for a while, and it seemed easy to configure and load data. Then I started creating projects and service assignment groups. After only a few hours of configurations (and glancing at any documentation), I had developed a working service catalog and service desk that included IT assets and CIs and major incident capabilities.

Jira Service Management incorporates elements from Agile methodology, like "start where you are" and offers out-of-the-box ITIL practices. It was apparent that Atlassian embraced a team-centric approach rather than a tool-focused solution.





## Atlassian Approach to Jira Service Management Upgrades

## Legacy ITSM Vendor Approach to System Upgrades

### Agile vs. Waterfall

Atlassian follows Agile methodology for Jira Service Management upgrades that optimizes flexibility, collaboration and efficiency.

Legacy ITSM vendors use a waterfall approach for upgrades that focuses on structured, sequential releases.

### Changes on your schedule vs Required yearly updates

Jira Service Management customers can control their rate of change by setting their environment's desired release track. Continuous-Products receive changes and features as soon as they become available. Bundled-Products receive changes as a group on the second Tuesday of each month

Legacy ITSM vendors generally provide software releases that contain new features twice a year.

### Ongoing feature releases vs. As-available bug fixes

Atlassian's approach provides customers with improved business value through shorter, predictable release cycles; smaller, higher-quality code sets; and more transparent, responsive feature delivery.

Legacy ITSM vendors use a waterfall approach for upgrades that focuses on structured, sequential releases.

### Loosely-coupled architecture vs. Tightly-coupled code sets

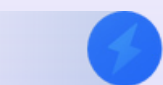
Jira Service Management is based on a loosely-coupled architecture where individual components are built independently from one another. Loosely- coupled applications allow teams to develop features that deploy and scale independently, which provides more efficient code delivery/upgrades.

Legacy ITSM system architectures are tightly coupled such that functions are rigid and interdependent. When one component changes, there is a 'domino' effect of more changes across the application which increases the overall code modifications and risk.

### True cloud product vs. Siloed SaaS offering

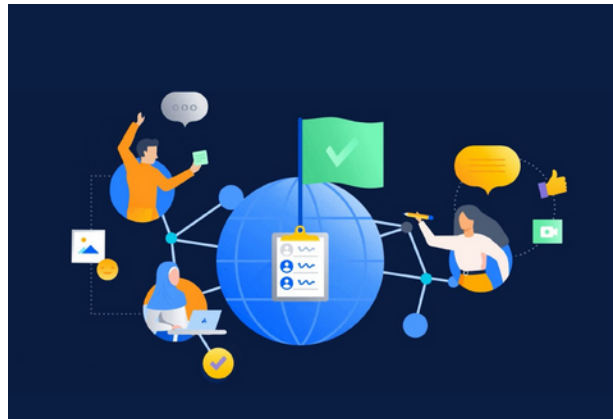
Jira Service Management is designed and created for Atlassian's cloud platform, so customers can scale confidently with embedded security, built-in compliance, unlimited instances, and simplified per-user licensing.

Legacy ITSM system are often designed to run on-premises, and vendor simply host the applications remotely. These are not true cloud solutions and can't provide you with the same benefits - scaling, upgrading, pricing, and security - as a true cloud system.



# A healthier approach to ITSM system upgrades

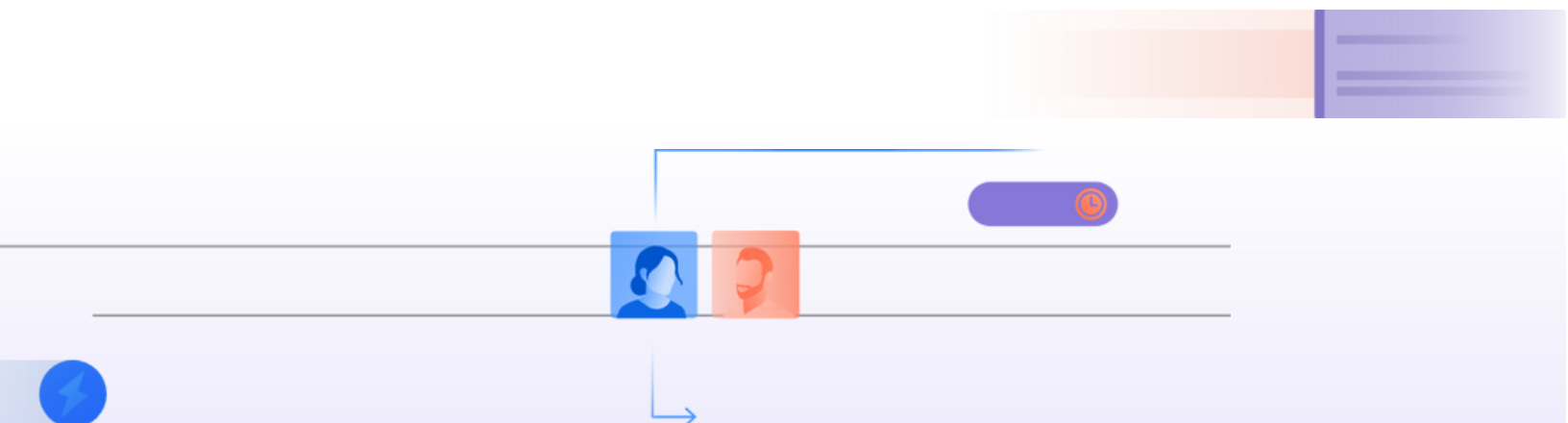
Getting into the issue of upgrades, Atlassian plans for technology evolution, so when new capabilities are available the functionality can be seamlessly introduced to the platform and products. Atlassian's cloud roadmap is public, so customers can always see what's coming. Jira Service Management's cloud-based system means that features are available for customers to consume continuously or on a scheduled basis.



## Customers can:

- 1 subscribe to weekly release notes for the most up-to-date notes on coming changes
- 2 control their rate of change by setting their production environment's release track (continuous and bundled)
- 3 create an isolated environment (aka Sandbox) to experiment with feature changes before hitting production
- 4 benefit from small, incremental updates that provide continuous improvement

Because Jira Service Management is based on loosely-coupled code sets, it provides improved code flexibility/reusability and more efficient application upgrades.



# Powered by the Atlassian Platform

Jira Service Management is built on Atlassian's unified cloud technology platform that powers open and efficient collaboration between teams, information, and workflows across your enterprise.



## ANALYTICS

Speed up decision making and gain insights across IT, Development, and external data sources. Visualize data with a low-code/no-code tool and template dashboards.



## AUTOMATION

Enable powerful cross-product automation without manual work. Easily orchestrate support, incident, and operations processes with rule-based workflow automation.



## COLLABORATION

Empower open collaboration and surface work across teams, not individuals. Create visibility through smart links, predictive cross-product search, mentions, and reactions.



## ADMINISTRATION

Access a centralized mission control across the Atlassian product portfolio to empower admins with advanced features for visibility, control, and ease of use.



## EXTENSIBILITY

Meet your enterprise's evolving needs by adding, integrating, customizing, and building new functionality with marketplace apps, our hosted app development platform, and integrations.



## DATA MANAGEMENT

Maintain control of your data across your Atlassian suite with a wide variety of data governance capabilities to ensure you meet the highest security and compliance requirements.



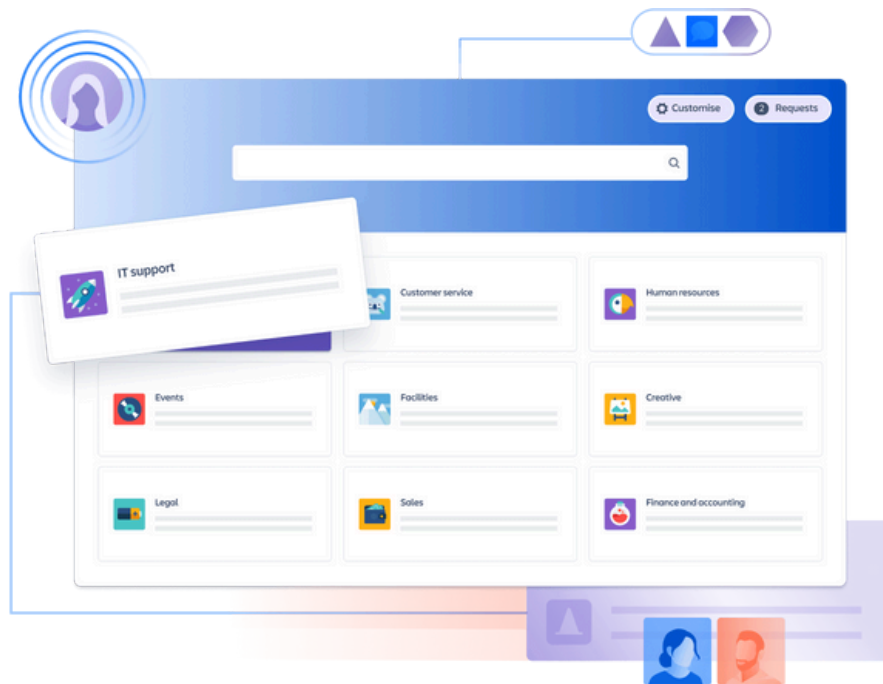
## INFRASTRUCTURE

Scale your service management with confidence on our world-class cloud infrastructure, built on a foundation of scale, performance, and reliability.



We all want to accelerate digital transformation so our organizations can “work smarter rather than harder”. I have found that Jira Service Management’s agile incremental change and improvement upgrade approach allows us to continue to validate releases and engage with end users – simply at a more continuous, faster, and non-disruptive pace.

If you’re embarking on a digital transformational journey, I recommend reviewing each application’s core architecture and implementation approach as well as the total cost of ownership. Consider your business initiatives that you are trying to achieve and choose a system that meets your needs and adds value to every customer interaction.



**Reach out to learn more about working with an Atlassian Solution Partner to reinvent your approach to service management.**

**Contact Jer-nee Consulting.**

**Partner name**

📍 Lehi, Utah 84043

✉ inquiry@jer-nee.com

🌐 www.jer-nee.com