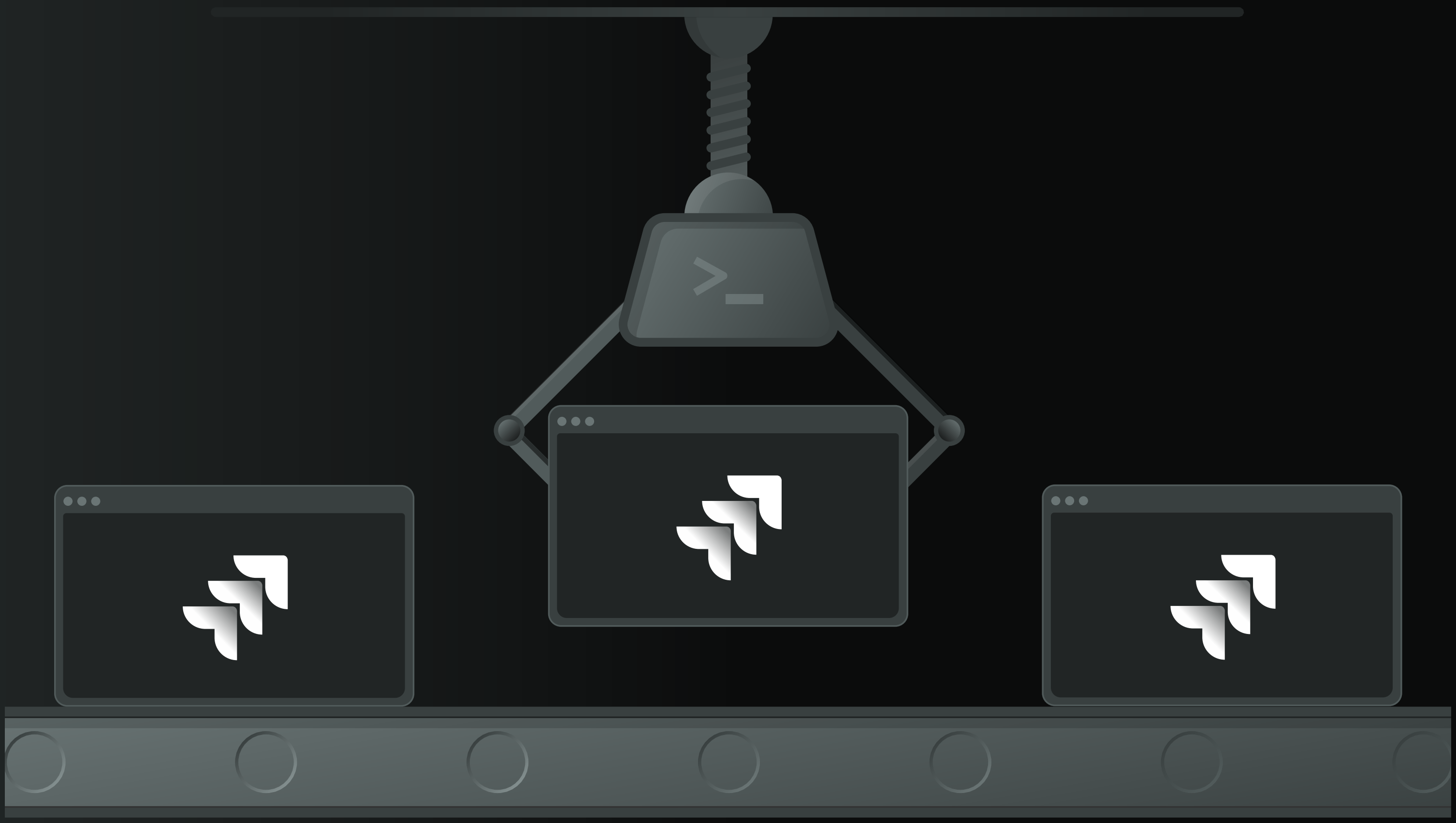
Workflow & Automation

appfire

Six automation strategies for Jira that'll make you (and your team) happy

A guide for better processes in Jira.



From Appfire
The #1 Connector App Vendor in the Atlassian Marketplace

Turn these strategies into action...
Inside you'll find implementation tips for your Jira Admin, so be sure to share!

Contents

Introduction

Your organization is scaling, your team is working with new people across multiple projects and a variety of instances. That's great news for company growth. But the bigger you get, the messier your processes become (unless you do some planning).

Messy business processes in Jira can lead to:

- Data duplication
- Human error
- Repetitive manual task creation and maintenance, and
- Inconsistent workflows between instances, projects and teams

In other words, major headaches for you!

Good news: we've consulted with experts to bring you **six valuable strategies for streamlining your business processes with automation**. Reclaim your team's time so they can do what they do best, instead of wasting time sorting through Jira messes and doing frustrating manual work.

If you lead a team that uses Jira, this guide is for you!

If you're managing an engineering organization, an IT service team or a business unit that uses Jira, automation (done right) can make your job much easier and increase your team's productivity.

And you don't have to be an automation expert to put these strategies to work: **just reach out to your Jira Admin for an assist**. We've included a few tips and links to resources so you can make these improvements happen.

In this guide, you'll find examples and use cases for specific types of teams — software development (for Jira Software), IT Service Management (for Jira Service Management), and business teams (for Jira Work Management) — but the strategies apply to all sorts of teams… Even teams that don't use Jira can benefit from some of these ideas!

Strategies we cover:

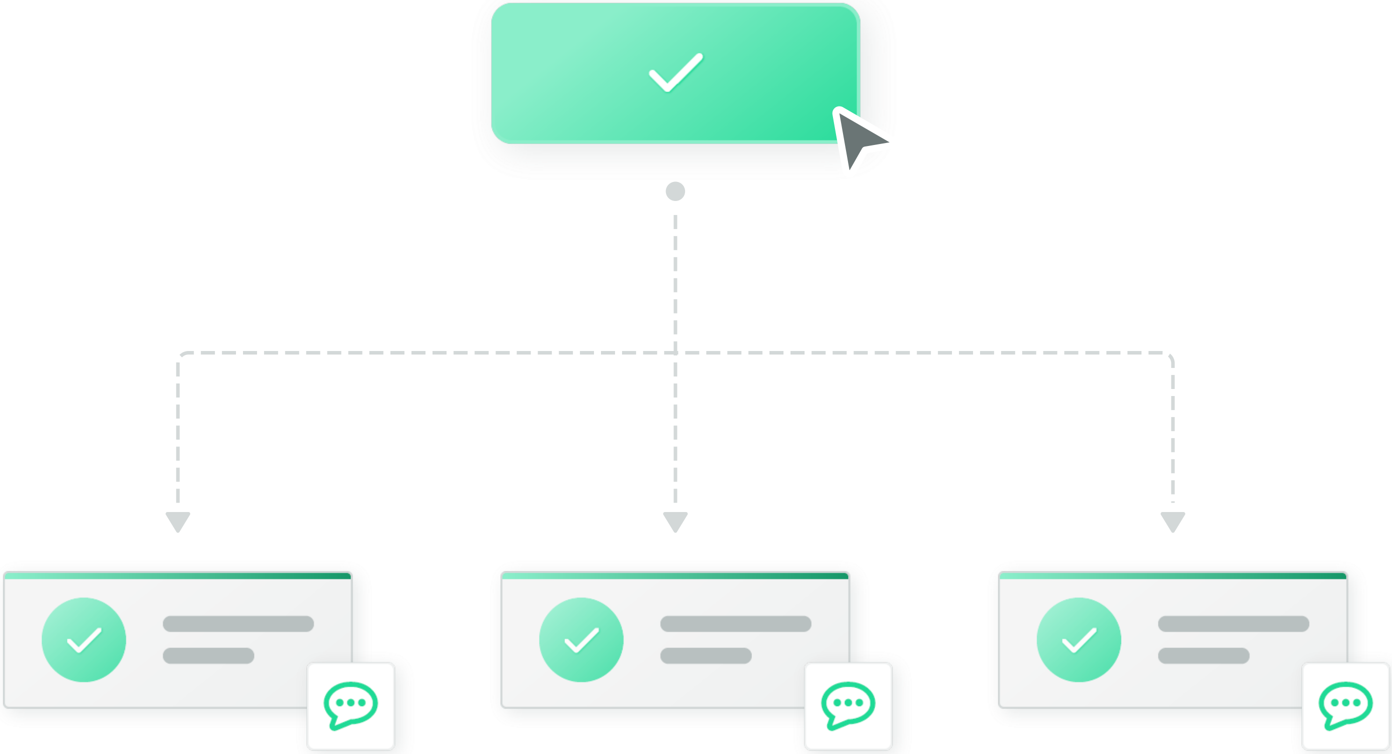
1. Empower human decisions with workflow automation
2. Create smarter business processes without complicating workflows
3. Streamline your Jira by keeping everything in sync
4. Improve communication with users and customers
5. Automate mundane tasks to remove user errors
6. Do more things in bulk

Bonus: Checklist for automating Jira

Read on for **strategies and tips** that will help you scale your team's expertise through automation. You don't often encounter a true win/win/win situation, but that's what automation offers.

- Your **team** wins when you increase productivity and job satisfaction by automating repetitive tasks that eat up their time.
- Your **customers** win when you use automation to reduce time to resolution.
- Your **company** wins when you reduce human error and keep your Jira instance up to date.

We've created this guide to help you win (and win and win) with automation.



Empower human decisions with workflow automation

1

Every team receives repeat requests: tickets or issues that require the same set of follow-up actions each time. Many of these tasks can be automated, freeing up team members to work on projects with more strategic value. (We talk about some of those below and in the sections to come.)

But in some cases, you need a human involved to decide on the correct course of action, then initiate the automated series. This is what's known as a "manual trigger."

For example, you might want a team leader to manually confirm before transitioning all issues relating to an old version of your software to "done," because that will trigger automated messages to everyone who submitted those issues.

Empowering human decisions with automation enables your team to automate tedious processes (like auto-creating tasks in different projects) while avoiding costly mistakes (like triggering messages prematurely).

Here are some ways you can empower your team's manual decisions with process automation using Jira.

Power up documentation updates for faster and easier release

For **Software Development teams** **Service Desk teams** **Business teams**

Empower your team to make the decision on documentation, then make sure nothing falls through the cracks by triggering tasks in a different project automatically.

New feature releases are part of any software dev's routine. Whether you're working on brand new features requested by customers (or your product team) or fixing bugs, these features need to be documented.

Even non-development teams — including service desks, marketing, finance, or HR — can benefit from documenting processes.

Major updates and new features may require a dedicated documentation page to explain the functionality. Improvements might just require adding some content to existing documentation pages. Small bug fixes might just go into the release notes, with no documentation changes needed.

Your team should decide which features need the full treatment and which just require a mention in the release notes. Automation can't make nuanced distinctions like that yet. But you can automate the actions that flow from that decision.

For example, automation can create a new issue in the documentation project to request a new page for the new feature (or request edits to an existing page, depending which path the user chose).

Since issues often contain useful information, the new documentation issue can automatically link back to the original issue for easy reference.

You can even copy version and other information from the original issue into the new one, even if the linked issue sits in another project. This will also help the technical writer to compile content for the new documentation page or update the existing one.

You save time and reduce the risk of human error because the user no longer needs to go into another project, create a documentation page, link it to the original feature issue and then provide all necessary information. A single click by the user starts this process, and the rest unfolds automatically.

Best practice advice

Review your current documentation project to see what updates are typically added as part of issue creation. Some of these items — like comments or reporters — can be automatically copied from the linked issue. (Reporters give technical writers a way to ask questions about features. Always useful!).

Using automation to create links and copy information between related issues simplifies processes for everyone involved.

How to set up this process

First, enable recording of the team member's decision: major update, significant improvement, or small fix.

Then you can trigger automation based on their selection.

EXAMPLE PROCESS:

1. Create a checkmark or drop-down issue field that becomes available after a certain transition, like when the feature transitions into QA status.
2. Prompt the user to select whether this feature requires documentation changes, either a new page or changes to the current page.
3. Enable the user to add comments as appropriate, so they can explain their reasoning.
4. The selection kicks off a pre-programmed series of **automated actions**:
    a. Create a new issue in the documentation project to request a new page for the new feature (or request edits to an existing page, depending which path the user chose).

b. Link back to the original issue for easy reference. Copy version and other information from the original issue into the new one.

But you *could* take automation a step further. With powerful workflow automation tools on the market, you can even create rules to publish release notes directly to Confluence pages. Check out our next example.

Tips for Jira Admins

Expert Tip from Atlassian expert and community leader Rachel Wright.

“Use the manual trigger to make repeated tasks easier. If a subtask or linked issue is needed, use automation to create the issue, assign it to the correct user, and fill in all the important fields in one click. You can even copy data from a parent or related issue, so the assignee doesn’t need to hunt for it in another issue. I love this trigger!”

Enable Jira users to build Confluence pages automatically — without leaving Jira

For **Software Development teams** **Service Desk teams** **Business teams**

Let team members work in their preferred tool instead of switching back and forth.

Leaving Jira to create a Confluence page on an issue reduces efficiency and increases the risk of human error, similar to what happens when users have to go to another project and create related tasks for documentation issues, as described above.

You can enable users to automatically generate a new Confluence page that includes all the necessary issue data and, of course, links back to that issue in Jira. You can simplify the process and ensure accuracy just by adding a button right in Jira that will transition the issue and kick off automation.

How to set up this process

First, enable recording of the team member's decision on which type of article: how-to guide, a new feature, or incident report.

Then you can trigger automation based on their selection.

EXAMPLE PROCESS:

1. Require the user to select in a custom field upon transition what kind of article they want to create. For example, a "how-to guide," "new feature," or "incident report."
2. Based on the user's selection, automatically place the issue into the appropriate project, and fetch the Confluence Page ID associated with that project. This ensures that the pages created all become "children" of that page.
3. You can then create a Confluence page dynamically using one of your template formats and inject values from within Jira (such as the description, the name of the reporter or issue, etc.).

With apps like Appfire's Power Scripts and Jira Command Line Interface (CLI), you can achieve this with a single line of code.

Tips for Jira Admins

Using third-party apps, such as Appfire's Jira Misc Workflow Extensions (JMWE), you could also have the Confluence page created as the individual Jira user (not as Jira itself), which eliminates the need to pass authentication back and forth. The created page is automatically linked back to the original Jira issue, completing the circle.

Resolve customer requests faster

For **Software Development teams** **Service Desk teams**

Assign customer tickets and bug reports to feature requests. Once a feature is implemented, all customer tickets will be automatically updated with comments.

In another section below, we'll discuss how to make sure that your customer-facing project — for customers to report feature requests or bugs, for example — stays in sync with your internal project.

For now, let's say a request comes into your service desk. The request might be an improvement to an existing feature, a report of a known or unknown bug, a new feature already on your roadmap, or a brand-new feature suggestion.

You'd want someone to add the request to the appropriate internal project, create a new issue (or add to an existing one), and make sure it all links to the original request in the customer-facing project.

After the issue's been fixed, you'd want to notify all affected customers. You can set up automation to update internal stakeholders and external users by email once the issue they were interested in has been released.

Best practice advice

As mentioned above, you *could* trigger an email. But you might prefer to have stricter control over the type of communication that goes out to customers so you can take a more personalized approach.

You might want to notify at least some customers manually. For that, you can use automation to create an internal comment that, in turn, triggers an automated message to your customer service representative. Then they can modify and customize the message before sending it to customers manually.

This process requires manual management, but the automatically triggered comments are key to ensuring that your customers are kept updated on changes they requested or bug fixes that impacted their instances.

In one of the later sections, we cover more ways to improve communication with your customers and users.

How to set up this process

First, enable recording of the team member's decision: creates a new issue or adds to an existing one.

Then you can trigger automation based on their selection.

EXAMPLE PROCESS:

1. Once the customer submits a feature request or a bug fix, a member of the team reviews and triages it.
2. The team lead managing these requests transitions the ticket to another status, such as "in review" (and even kicks off an automated message to the customer).
3. As part of this transition, the team lead adds the request to the appropriate internal project: creates a new issue, or adds it to an existing one.
4. The process automatically creates a link from the customer ticket to the issue in the selected project for easy reference.
5. The new issue (with a feature request) will maintain links to any other customer requests that come in requesting the same feature. This essential information can help you see the volume of requests for each feature. (BONUS: You'll get a better understanding of your customer's needs because comments can be easily accessible.)
6. Now you're able to notify all affected customers once the issue that they were interested in has been resolved.
7. Using automation, you can add a message for impacted customers in the comments section of their ticket and possibly trigger an email.

You can use Appfire's JSU, JMWE, Power Scripts, and Jira Command Line Interface (CLI) apps to empower automations in the above section.

Create smarter business processes without complicating workflows

2

Jira is a powerful tool, but you need to configure it properly to set users up for success. If your process heavily relies on data entry or manual calculations by end users, you're inviting mistakes.

Automating repetitive data entry tasks minimizes the opportunity for human error. This type of automation also improves your team's productivity, freeing them up for more challenging work.

This adds tremendous value to your business because the results you achieve impact other teams' projects, budgets, and processes.

In this section, you'll learn how to keep your projects running smoothly by automating select parts of your processes. You might build automations related to employee onboarding, for example. Or you could create tasks users do on a weekly or monthly basis. You might create automations that streamline your approval processes. You could even automate calculations.

Save hours by automating repeat tasks

For **Software Development teams** **Service Desk teams** **Business teams**

Users often need to create the same tasks and subtasks for different epics. You'll see this when creating things like QA checklists, new feature requirements, and approval processes.

Ideally, you want to set up repeat tasks only once. Even if the initial set-up takes a while, you'll save time in the long run by running the automation over and over again. Take the long view to set your team up for long-term success.

Make templates for repeat tasks like employee onboarding

Automating repetitive tasks in your employee onboarding will save your team hours of valuable time. When a hiring manager creates an on-boarding request, you need to make sure that certain tasks are completed before the new employee starts. For instance, you need to request a laptop and phone, and set up an email address for the new hire.

This same opportunity presents itself in other areas, like product announcements or marketing promotions: you need to complete several of the same tasks over and over.

Any Jira admin or power user has, at some point, been asked: "How can I create a template of issues?"

Depending on requirements, implementing a solution for this may take anywhere from five minutes to a couple of days (or weeks, or months). Using apps can help you to stay on the short end of that range.

Tips for Jira Admins

Damian Rosochacki of Isos Technology

"Tools like Jira Miscellaneous Workflow Extension (JMWE) app can help you quickly set up templates for epics, issues, and subtasks in less than five minutes." See how to set up templates.

Create repeatable weekly or monthly tasks

Another approach is to create repeatable tasks according to your team's work schedule. For example, help teams automatically generate recurring tasks at the start of each week. You might use automation to create a weekly schedule for security checklists or team reviews, or a monthly schedule for performance reports.

You could even create a rule that sets the due date for these kinds of repeat tasks. For example, you might set the due date to the 10th of the current month if the issue is created before the 5th, and to the 25th of the current month if the issue is created after the 5th. You can set up whatever rules work for your team.

Jira Admins can see how it can be done using JMWE's Scheduled Actions or Power Scripts' SIL Scheduler.

Automate approvals without complicating Jira workflows

For **Software Development teams** **Service Desk teams** **Business teams**

Generally speaking, it's best to avoid overloading your Jira workflows with too many rules and complex screens. Create screens that are easy to work with, keep templates simple and flexible, and make everything easy to adjust.

To simplify their work, you might give users multiple ways to transition issues. Maybe enable them to skip statuses when needed. But sometimes, skipping a status is a bad idea. It could, in some situations, create a compliance problem or other issue.

For example, take an approval status. You might fail an audit if work was started on an issue (or an issue was completed) before it was even approved.

But you can automatically prevent that from happening. You can even automate approval processes to ensure separation of duties for SAS-70 compliance and make sure the same user can't trigger two incompatible transitions.

All you need is an app and a little configuration.

How to set up this process

First, add and configure a workflow validator to support automation for the approval process.

Next, set up better screens for users, reports and dashboards.

EXAMPLE APPROVAL PROCESS:

1. Set up a validator to ensure that an issue passes through the “approval” status before work is started or before an issue reaches its final workflow status.
2. Program conditional approval to allow select issues to bypass the “approval” status in certain situations. For example, if approval is required depending on the cost of the project/equipment etc., set conditions: anything under $500 can bypass approval, anything $500 or over requires approval, and anything $2000 or over requires approval of a supervisor.
3. Record additional details during the approval process, such as the name of the approver and time stamp.
4. Display approval details on users’ screens and use them in reports later. You can also display things like who transitioned an issue, and display transitions in issue activity views for end-users.

See how to set this up using JMWE

For Jira Admins: Take a look at how you can automate project approval using Power Scripts.

Tips for Jira Admins

Skip approval validation while cloning an issue

If you set up validation processes — and that is strongly encouraged — you ensure better workflow.

You can validate fields value on input — either during a transition or outside of a workflow transition — to make sure that users enter information correctly.

But the need to validate a workflow transition might vary depending on whether you are creating a new issue or simply cloning one. If validation is not required for cloned issues, you can easily skip it when cloning an issue. Apps like JMWE and Power Scripts handle this through conditional validation.

Automate expense reports and budget tracking

For **Software Development teams** **Service Desk teams** **Business teams**

You've just returned from a work trip to kick off a project with your new client. The kickoff went well, and the client is happy, but you're exhausted from the long flight. The last thing you want to do is complete a lengthy spreadsheet so the finance team will reimburse your travel expenses. Sound familiar?

Rachel Wright, Jira expert and author of the *Jira Strategy Admin Workbook*, knows the feeling. She dreads filling out expense reports, so she came up with an easy way to submit them in Jira instead.

How to set up this process

Rachel's recipe for painless expense reports:

1. To make the expense reporting process easier, configure a Jira project for reimbursement tracking and install the JSU Automation Suite for Jira Workflows app to add calculation abilities and extend workflow functions.
2. Create a custom issue type, workflow, and custom fields to collect and calculate trip details, mileage, and costs.
3. Use subtasks to track reimbursement progress and epics to group expenses together by type.
4. Now the leadership team can easily see how much money is spent on client travel.

In this video, you'll get an in-depth look at how to automate this process in Jira with JSU Automation Suite for Jira Workflows.

Avoid mistakes by automating calculations

For **Software Development teams** **Service Desk teams** **Business teams**

Have you ever wanted to automate calculations based on data from your Jira fields? Jira allows you to create custom fields, but it doesn't provide a way to perform any calculations within those fields. That's not ideal because automating calculations saves time and reduces errors.

Fortunately, you can automate calculations: just not using Jira's native functionality.

Say you want to **calculate labor cost estimates** based on custom fields that record "work hours" and "hourly fee." In each issue, you want to display a result of a fairly straightforward calculation of the labor cost of the project ("work hours" multiplied by "hourly fee").

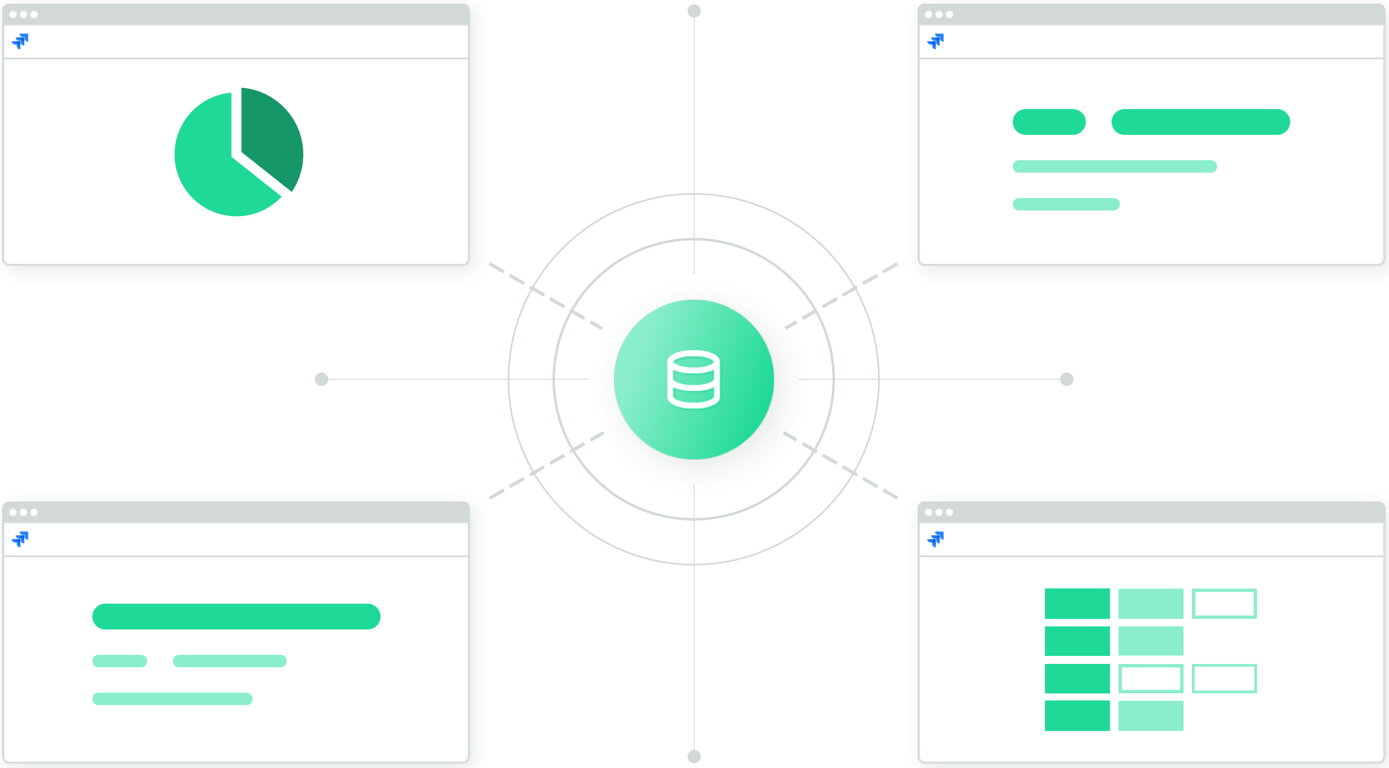
Although this can't be automatically done using Jira, you can easily automate that calculation with third-party apps like Appfire's JSU, JMWE, and Power Scripts. Automating this improves accuracy and eliminates the need for manual calculations.

Better still, automated calculations will automatically adjust if a custom field value that drives the calculation changes. In the above example, if the number of hours worked increases or the hourly fee changes, so will the labor cost calculation.

Similarly, agile teams might **automate adding up story points for subtasks in a parent story**. This way, the parent story always shows the correct total of story points. Because the total will be automatically updated if the story points for one of the subtasks change.

Jira Admins can see how it's done using JSU, JMWE or Power Scripts.

You can use Appfire's JSU, JMWE, and Power Scripts to empower automations in the above section.



Streamline your Jira by keeping everything in sync

One common issue is updating a single data point that appears in multiple places. If you've ever changed your address or received a replacement credit card, you know that even a simple change can take a lot of time when you need to update information in dozens of different systems.

Your team experiences that same challenge daily: small, simple changes made exponentially more time-consuming because they must be repeated in multiple processes, documents, systems, etc.

Say you want to make sure all issues in the current open sprint are transitioned to "done" when work is complete. (People sometimes forget that last step.) A team member would need to manually check the status of every issue. And you'd have to check periodically to make sure that those updates are actually happening.

Process owners have more strategic work to focus on, but keeping your Jira instance clean is critical to ensuring successful process implementation. That's where automation comes in.

In this section, you'll learn how you can set up processes to keep projects aligned and in sync.

Keep things aligned by automatically closing issues

For **Software Development teams** **Service Desk teams** **Business teams**

When issue status does not reflect reality, reporting is a nightmare.

The most common use case for the synchronization of projects is this: child issues are closed, but nobody remembers to close the parent issue. Users have moved on to the next project, and the derelict parent issue sits in its incorrect open status forever.

You need to close the parent issue when all child issues are closed. Or maybe you have the reverse problem: your epic is closed, so you need to close all associated stories, bugs, and tasks that were forgotten.

You want to keep things synchronized. This simplifies reporting and ensures a much better Jira experience. Good news: there are plenty of automation options that can help.

Automatically close issues for straightforward reporting

All completed or unneeded issues should be transitioned to their final status. Take this step off your user's "to do" list: automatically close parent issues once all child issues in the project are resolved.

Don't make users manually transition the parent issue or you risk having an instance full of old, orphaned issues.

Tools like JSU, JMWE, and Power Scripts by Appfire can help your team to get this done.

Tips for Jira Admins

Do your epics stay on the backlog of your Jira agile board after closing them?

That's because, in addition to the workflow status, epics also have a field called "epic status" that's only accessible on the backlog screen.

Only epics with the field epic status marked "done" get removed from the backlog. You can easily automate the removal of closed epics from the backlog.

See how it can be done using JSU.

Other reasons you might want to automatically close — or reopen — issues

Consider all your current processes candidates for this type of automation. For example, you can create automation to transition an epic to "in progress" status when at least one of the stories is changed to "in progress."

And, as mentioned above, you can automatically close parent issues if all children are closed (or vice versa) to keep things synchronized.

But you might also consider automatically closing - or reopening - issues for other reasons. Here are some ideas:

- You can automatically close customer tickets if no reply has been received after 5 days.
- If a customer comments on their closed request saying the situation wasn't resolved to their satisfaction, an automation rule can reopen the original request instead of creating a new one.

- A previously-closed issue can be automatically reopened if a customer, a reporter, a department lead, or a member of a predefined group (like your executive team) creates a comment.

- You can set up an automation rule to analyze the latest comment and auto-resolve the issue if the words like "complete" or "resolved" are found.

- You can automatically resolve all issues in the current open sprint that are awaiting deployment by using automation to transition issues to "done."

- You can use an automation webhook. Many software projects have some type of automated build and deploy tool (like Bitbucket) that automates the building, testing and deployment of code.

Synchronize software versions across internal and external projects

For **Software Development teams** **Service Desk teams** **Business teams**

In one of the sections above, we discussed how to ensure customers are notified about changes to their submitted feature requests. But that's just part of the story. What about doing the work vs. communicating the status?

For software projects, you'll typically want two projects: one public, for customers to report bugs and request features for your newly released (or updated) software; and one for your development team to track their work on these issues.

With this two-project set-up, your team can triage requests before opening issues to work on so your Jira doesn't get slammed.

But for this to work, you need to keep your software versions in sync for both projects. That way, you know customers are reporting issues on the right version.

With automation, that bit's easy. You create a rule: anytime you release a new version of the software in your team's internal project, the system automatically creates a corresponding version in the public project at the same time.

Keep your projects in sync so that issues created track to the correct version of your software.

Automatically add high-priority bugs to current sprints

For **Software Development teams** **Service Desk teams**

If you have a process for creating high-priority bugs, it probably involves having support staff review high-priority bugs submitted by your top customers.

Once your team verifies the bug (or raises another bug to high-priority status), you'll want to keep these bugs synchronized, and link them between your team's internal project and your public-facing project. Most importantly, you might want them added to the current development sprint.

Then you can use global rules to synchronize and link these high-priority bugs automatically.

Still, manually triaging these bugs in the first place takes effort.

If verification isn't critical for bugs submitted by top customers, you can create rules to automatically synchronize and link high-priority bugs submitted by certain customers. (You can base this on predefined classifiers located in another field, like revenue numbers, or use other applicable rules.)

This will automatically change the scope of your current development sprint.

Take your synchronization processes to the next level

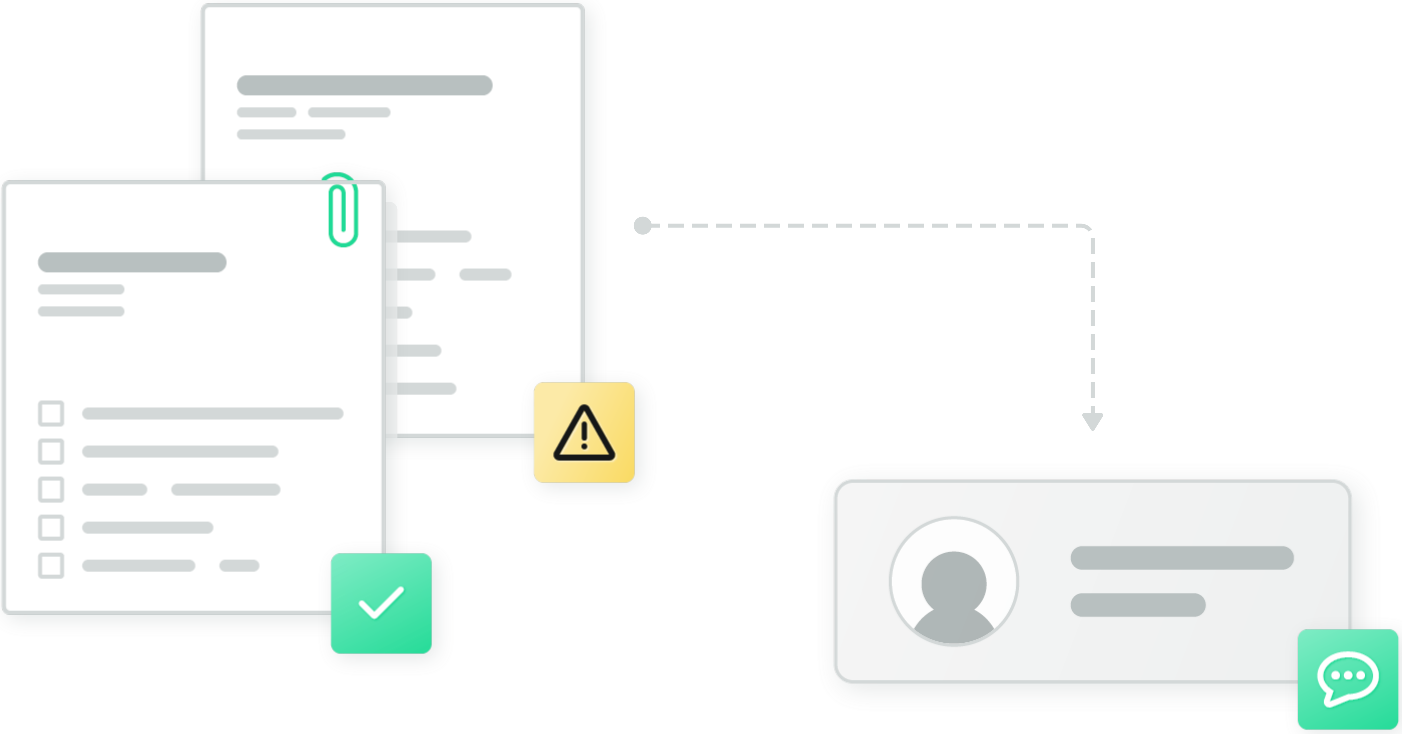
For **Software Development teams**

Once you've set up some common process automations described above, you might consider some more sophisticated rules.

For example:

- If you track estimates of an epic, update the original estimate whenever tracking fields in the epic's underlying issues change. This keeps estimates for an epic in sync with its child issues.
- When starting a sprint, update all stories within that sprint to set the due date to the end date of the sprint. And set the fix version to the next unreleased version in the project.
- Set up automation to update version releases for issues in a current sprint: once released, automatically update the release version for all related tasks to keep everything in sync.
- If code review is rejected, automatically transition the epic — and all related child issues — back to the previous status, or to a particular status identified in your development cycle. This ensures that important actions like code review get consistent handling.

You can use Appfire's JSU, JMWE, and Power Scripts to empower automations in the above section.



Improve communication with users and customers

4

According to Salesforce research, 86% of employees and executives cite a lack of effective collaboration and communication as the main causes of workplace failures.

That’s why change management is so critical. To perform effectively, staff need access to accurate, up-to-date process information.

You need personalized notifications (and several reminders) to staff about changes to procedure documents, and you’ll want to ensure that everyone is using the most current version. One way to do this would be to create a personal dashboard that triggers emails to a dynamic list of affected stakeholders.

Jira can send notifications to users via email when specified events take place, like when an issue has been updated or a comment has been added. You can also send emails to various users containing the results of saved filters, or send results on a scheduled basis.

With Jira Service Management, you can create custom templates for sending notifications to customers via email in place of standard event notifications. You can send emails to any email address, user group, or list of users whenever the rule is activated.

Here are some ways you can improve communication processes in Jira and beyond.

Improve flow by displaying custom messages at the right time

For **Software Development teams** **Service Desk teams** **Business teams**

Effective communication is all about getting the right information to the right people at the right time. One simple way to reduce human error in the handling of any issue is to display a custom message during a transition. Remind users about common tasks people often forget at this stage in the process.

One option is to set up a required field with predefined values so that you can automate certain processes. This helps to ensure compliance but, from a process perspective, it's not as simple as just selecting an option to display.

In some cases, you might want to create different paths for selected options. For example, one option from a dropdown might kick off a transition, and the other one might require a different action (like adding a document as an attachment, or emailing a form to users).

You can set up automation processes to handle these choices differently (we describe some of these options in the following section), but the important thing is to make these processes clear for users. Consider upping your "notification" game — display messages to users only when required, and in different steps of the process.

Here are two examples of how displaying messages to users at the right time makes a big difference:

Display a message to a user while requiring an attachment, conditionally

In Jira Service Management, a customer request must have either a custom field or an attachment. You can set up the process to require an attachment conditionally — *only* if the custom field is left empty. So if the service desk representative does not enter a value in a required field, a system prompt warns the user of a missing required attachment.

Here's one way to set up a process like that using JMWE by Appfire.

Display a message to a user while validating input on field changes

You can validate a field change when an issue hasn't transitioned, and display a message to users. This feature simplifies automation outside of Jira workflows. This video demonstrates how you can use JMWE's Event-based Actions to set this up. And in this Power Scripts Master Class lesson, you can learn how to control fields in a given Jira project. Using Live Fields in Power Scripts, you can show or hide custom views on an issue screen, set values, manipulate the options in select lists, and more.

Enable agents with canned responses and automate actions

For **Service Desk teams**

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. In another section below, 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 in 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 the 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 and pasting also makes your agents happier, so they're less likely to quit. With automation, they spend less time on repetitive tasks, freeing them up to focus on more creative, satisfying work.

Best practice advice

Third-party apps like Canned Responses by Appfire enable your team to create dynamic templates and customizable signatures.

These templates can also trigger an automatic sequence of actions without tricky workflow configuration. Thanking a customer, for example, can automatically close the issue and transition the status to "done." Similarly, sending an escalation template can automatically assign the ticket to the correct team, streamlining internal workflows.

With Canned Responses, your agents can create a delayed message or schedule a comment. 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. 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 time, achieve higher productivity, and improve their time to first response ("TFR").

Take your communication to the next level

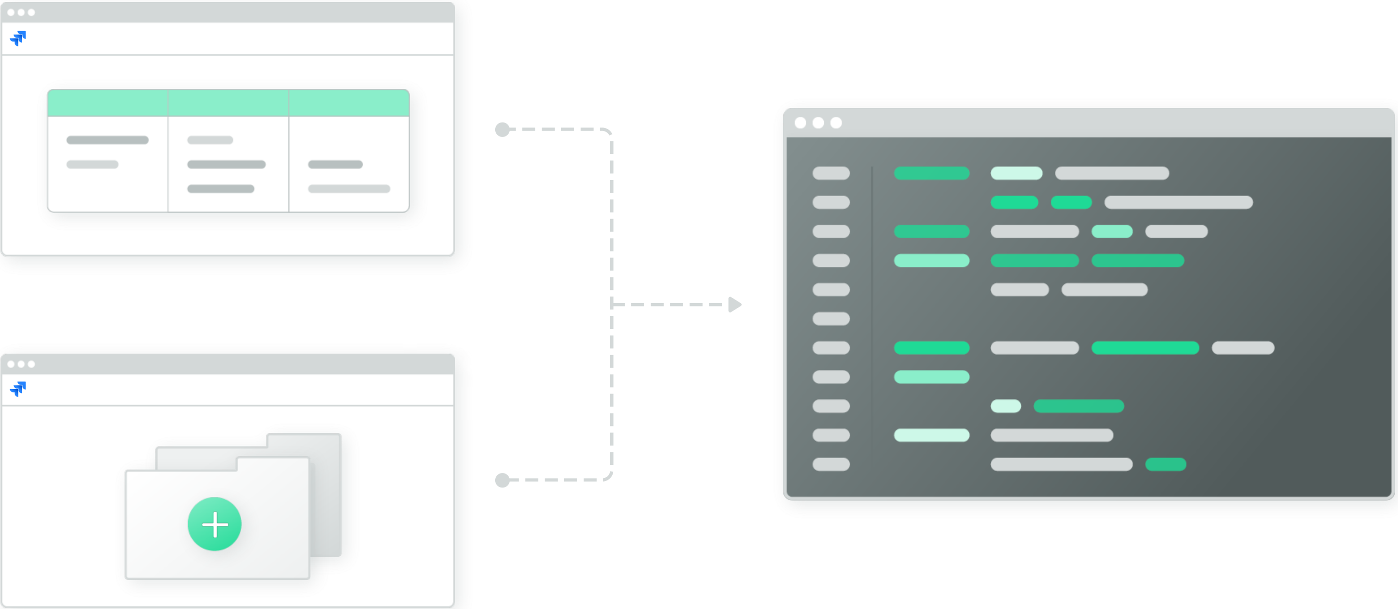
For **Software Development teams** **Service Desk teams** **Business teams**

Once you've set up some of the common communication automations described above, you could consider doing even more to improve your business processes.

For example:

- Automatically notify stakeholders — everyone involved in the sprint — of any scope changes to active sprints. The rule will automatically monitor for any change in the sprint fields of any related issue, and trigger a notification to team members.
- Set up automation to notify the legal team to update your terms of service when features of the current sprint are released.
- During an outage, immediately notify customer service to be ready for a large volume of requests.
- Send reminders and alerts for overdue tasks. No one likes to be overwhelmed by too many reminders. But you can strike the right balance by using third-party apps to create better communication rules, including action-based and schedule-based.
- Notify customers when a new version is released. List the items that were fixed or added in the newly released version, and check all tickets marked with certain issues to let people know that their issue was fixed with this update.
- When a comment is added that includes words like "important" or "urgent," automatically notify everyone in the group (including the assignee)

You can use Appfire's JSU, JMWE, Power Scripts and Canned Responses to empower automations in the above section.



Automate mundane tasks to remove user errors

5

User errors cost time and money. Bad data hinders business processes and makes reports practically useless. But taking time to fix and validate your data takes resources and can cause overall productivity to go down.

So how do you keep your data and processes clean? By preventing errors in the first place.

Automating mundane tasks minimizes human error and makes your team happy. Here are just a few ways you can reduce the opportunity for mistakes (and reclaim your team's time) through automation.

Prevent mistakes by automating actions based on field selection

For **Software Development teams** **Service Desk teams** **Business teams**

When a help desk technician triages customer support requests submitted through the Jira Service Management customer portal, they have an opportunity to classify requests in order to take different actions based on those selections.

For example, for some types of requests, tickets might need to be escalated immediately. For some others, you might want to trigger automated responses for certain team members. Options are limitless.

But if you use the drop down field to list all available options, that list will be incredibly long, and a technician could easily choose the wrong classification. Your best bet is to create **two fields**, where the first selection determines the second field’s available options.

Then, you can use different routes for different automations and avoid classification errors (always a good thing).

How to set up this process

First, create two fields where the first selection determines the second field’s available options.

For example, say a help desk technician receives a request and classifies it by choosing a “component” representing an impacted software application or business area. They use a second field to select the type of support needed.

Then you can trigger automation based on their second selection.

1. Set up component-specific options that display the type of support needed. For example, if the component is “email,” the only applicable support selections are “troubleshooting” and “monitoring.” If the component is “network,” the only suitable support selections are “monitoring” and “security.”
2. Set up automation with different actions for each. For instance, if the component is “network” and the support option is “security” the issue can automatically be escalated to the security team as high priority.

Jira Admins can see how it’s done here using Appfire’s JMWE.

Reduce workload by copying attachments from one issue to another

For **Software Development teams** **Service Desk teams** **Business teams**

Many tasks and projects in your Jira contain important information you need to keep visible in related tasks.

For example, say you task your team with creating brand guidelines for the organization. You've uploaded a creative brief PDF document into a Jira issue as an attachment. You've also created subtasks for each team member to work on.

These issues are linked, of course. But viewing the creative brief in the parent issue requires multiple clicks for your team members. To make sure everyone has easy access to the brief, you'll want to attach it to all subtasks.

But you don't have to do this manually.

Automation can help you make any document accessible to different users working on tasks, regardless of their permissions to see parent issues.

You can use automation to copy the document from the parent issue to your team's subtasks. And every time a new subtask is created in that same project, the file will be added automatically.

Jira Admins can see how it's done here using Appfire's JSU, but several automation apps offer functionality to help you copy/move attachments automatically.

Tips for Jira Admins

Do more work in less time with powerful scripting: Copy or move multiple attachments at once

Take a look how you can get that done with just a single command using Atlassian Command Line Interface (CLI) by Appfire. (You'll also find several other tips that'll make your job simpler!)

Alleviate bottlenecks by automating ticket assigning and routing

For **Software Development teams** **Service Desk teams** **Business teams**

Automating ticket assignments, applying priority level, or automatically entering information in important fields like due dates improves efficiency and drives down costs. In IT service management, for example, automating tasks related to triaging requests reduces the cost per ticket dramatically.

If you have a service request catalog that clearly identifies the types of services your Service Desk, HR, or Marketing teams offer, you can use these categories to triage inbound requests.

Above, we covered **how to classify requests based on components** to narrow down the categorization. This way, service department tickets can be automatically assigned to the right service queues, approval processes, and implementation tasks.

But many teams also use email requests to create service desk requisitions, and these typically end up in a service queue that requires human intervention to triage and categorize them.

Instead of creating a bottleneck right at the start, use automation to shorten the response time. You can automatically categorize incoming email with the correct component based on the presence of specific keywords. Then, automatically route and assign the requests to the correct team.

And that's just one example. You can auto-assign and automate processes to kick off different workflows based on all sorts of criteria. For instance, with an app like Power Scripts, you can auto-assign tickets based on current workload. So if Agent A has 20 open tickets and Agent B has 10, a new ticket would get assigned to Agent B because she has fewer open tickets.

Take your productivity to the next level

For **Software Development teams** **Service Desk teams** **Business teams**

Here are some examples of processes that can supercharge your team's productivity:

- Use the project and component selected to kick off different workflows. For example, you can automatically assign any requests that are infrastructure-related to a member of the infrastructure team.
- Automatically assign Issues created by upper management as highest priority.
- Assign issues to specific users or groups of users based on current workload or according to a pre-defined order (i.e. whoever's up next gets the new issue).
- Use custom field data to assign someone who has a certain skill or office location.
- Automatically escalate requests to the correct team when an issue field such as due date, priority level, or any other important field changes.

These automations will shorten the response time for ticket handling and increase team productivity.

Tips for Jira Admins

From Atlassian expert and Appfire Product Manager Jonathan Muse

"People tend to underestimate how much time they can save automating small everyday tasks. They focus on the big problems and ignore the small things that are so simple to automate. But even small, frequently recurring tasks add up to days, weeks and months.

So I always recommend starting small – talk with the team to understand the most frequently recurring tasks and start from there."

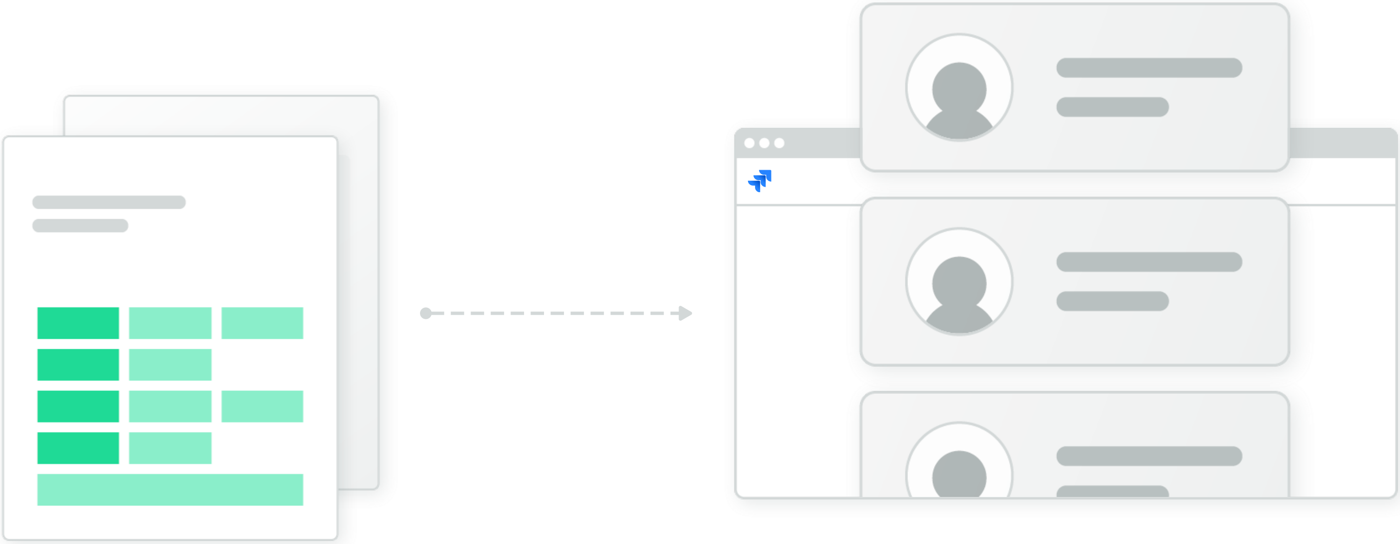
Best practice advice

YOU CAN DO EVEN MORE. JUST ASK YOUR JIRA ADMIN.
Below we cover some of the examples and automation hacks Jira Admins use to shave hours off their workday.

Like doing things in bulk. Done manually, individual tasks can take hours or even days to do. But these can be easily automated as bulk actions using apps, and often completed in just an hour or two.

Read the next section and discover how to automate some things you might not have thought of.

You can use Appfire's JSU, JMWE, and Power Scripts to empower automations in the above section.



Do more things in bulk

6

Some tedious, repetitive tasks are unavoidable. Like taking out the trash. Unless you have someone else in the house who'll do it, you're stuck bringing those bags to the curb.

Talking shop, there are plenty of tedious one-off tasks your team has to handle. Correcting an error or creating hundreds of users or tasks in one shot would be time-consuming and tedious if you had to do it manually, one update at a time. You might even make some mistakes (because who wouldn't in that situation).

But you *can* automate a surprising lot of one-off tasks if your team uses Jira. A few simple automations can help your team reclaim hours of time each week. Jira Admins can help with that. You just need to ask.

While these automations should be tailored to your business needs, below, we share some ideas inspired by the Appfire community, along with tips on how each can be done, and why.

And speaking of tips, we have a checklist for your Jira Admin included in this guide, so don't miss that.

Here is how you can achieve more by doing things in bulk with one-time automations.

Standardize processes by bulk-cloning and customizing issues

For **Software Development teams** **Service Desk teams** **Business teams**

Some organizations have multiple software development teams working on similar features, just for a different set of requirements or at different times.

For example, one of your teams might work on software features for iOS to support Apple mobile users, while the other team works on those same features for Android.

Say that the iOS team leads the way when it comes to the development of a new feature. Once the feature is launched, the Android team needs to replicate the process for most of the tasks.

This maintains consistency between projects and processes.

Luckily, Jira admins can clone up to 1000 issues at once. In this example, your Android team could get a head start by bulk cloning the iOS team's relevant issues into an Android project.

As part of the bulk cloning process, it's important that you get to decide and adjust what gets copied from the original issue into the cloned one. This ultimately reduces redundancy and saves time.

Jira admins can easily bulk clone multiple issues from the same project or different projects. See how to set this up with Clone Plus for Jira by Appfire.

But what if some of your data comes from external sources? In this case, Jira Command Line Interface (CLI) can help you make this process more smooth and efficient. Here's how one of Appfire's customers solved this issue.

And if your work involves a lot of repeatable processes, scripting-based add-ons like Power Scripts and Jira Command Line Interface (CLI) can help you streamline things. Because you can easily reuse previously created commands or scripts.

Skip the data entry and add new users from a spreadsheet in bulk

For **Software Development teams** **Service Desk teams** **Business teams**

Maybe you merged with another team, or your company acquired another company, or your team's just started using Jira. You need to add a large number of users to your Jira account, and fast. Instead of doing this manually, you can automate the creation of new users in Jira from a spreadsheet.

Apps like Jira Command Line Interface (CLI) by Appfire can help you bulk-upload hundreds of users from a CSV file in minutes.

And that's just the beginning. You could take automation even further. With sophisticated scripting apps, your Jira Admins can get work done faster, taking bulk-actions with conditional logic within large Jira instances.

Tips for Jira Admins

From Atlassian expert and Appfire Product Manager Michael Kuhl

Forget making updates one at a time: update issues in bulk

"When designing workflows, it's important that all statuses and transitions have a name and a purpose that describe the next action for the assignee rather than an attribute of the work in progress.

For example, rather than having 'Start QA' and 'QA Finished' statuses, use one 'QA' status. This conveys to the assignee and anyone else viewing the issue that Quality Assurance is the current activity.

Use the 'Start QA' and 'QA Finished' text for the transition names instead. That way, when someone's working on an issue that is in, say, 'Development' status, choosing the 'Start QA' transition from the 'Development' status clearly conveys the intention that development is done and QA should start.

This makes the next action clear to the assignee and the viewer."

Jira Admin’s opportunities for automation

Every job includes some tedious, repetitive tasks. Well, most jobs. But not yours. Not anymore.

Because you’re about to see how you can automate several important (but tedious) manual tasks in Jira.

We asked industry experts and members of the Appfire community to share some of their most valuable automations, and now we’re sharing their top four with you:

1. Automation that triggers an email if there’s an app outage. This gives you a heads-up on potential performance issues.
2. Integrations via RestAPI and webhooks that connect external tools with Jira so you can import/export data. Take a look at how Power Scripts can help with this.
3. In Jira Service Management, integrate with the user directory service to automatically provision new accounts based on service requests.
4. Enable faster password resets to Jira (and other) instances or, even better, reset passwords automatically!

Tips for Jira Admins

When you set up automation for repeat tasks, consider using JMWE’s Shared Actions feature. It enables you to create ‘shared’ JMWE post functions (or sequences of post functions) that can be reused in multiple workflow transitions.

For example, you can configure a post function to send an email, then reuse it in different workflows.

You can use Appfire’s JSU, JMWE, Power Scripts and Jira Command Line Interface (CLI) to empower automations in the above section.

BONUS

Checklist for automating Jira

Automation can help you streamline your business and make your processes more efficient. But to create the right rules and get the most value from them, you need a plan. Follow these steps to get started.

□ **Look for patterns in existing requests.** Identify opportunities to improve efficiency.

□ **Engage with users.** They can provide valuable insights into which tasks take the most time and how you could use automation to increase productivity.

□ **Plan your rules.** Any new rules should align with your larger business and process goals.

□ **Scope your rules.** Consider whether the rules you're creating are for a single project, multiple projects, or for your organization as a whole. Multiple projects often share workflows and configurations. If that's the case, automating workflows that apply to those rules makes things easier for your team because you can maintain a single rule instead of having to update duplicates.

□ **Assess feature requirements.** If native Jira doesn't support your requirements, check out any third-party apps that are already available in your Jira instance. Or research new apps using the Atlassian Marketplace or hub.appfire.com.

□ **Design the automation rule.** Try to avoid creating rules that are overly complex. Even when scripting is involved, make the rules as easy as possible for others to understand. Bonus points if you create rules that can be easily reused in similar situations.

□ **Test your rule on a single Jira issue** before increasing the scope of issues impacted. Consider triggering the rule manually to make sure it operates as expected before having it run automatically.

□ **Use the "Log action" feature** to capture any outputs and make sure the correct information is returned or calculated.

□ **Document your rules.** Include your business case for each rule, which should align with your organization's business goals. Proper documentation will help you create repeatable, standardized processes.

□ **Document your apps**. Specify which apps you're using for Jira workflows and automation (and what you're using them for). Include links to documentation and help desk.

□ **Keep a copy of your rules in a version control system.** You'll want to maintain backups of each rule, and keep track of the change history. Being able to roll back quickly when someone changes a rule and creates errors will help you avoid major problem.

Tips for Jira Admins

Bonus Tip from Atlassian expert and community leader Rachel Wright

“The manual automation trigger is a Jira administrator’s best friend! I use it to test new automation conditions and actions I’ve created. This way, if there’s a bug or the automation doesn’t work the way I expected, only one Jira issue is impacted. Once the rule is ready to use, just replace the manual trigger with an automatic one. Safe and easy!”

Appfire apps that help Jira admins get this done

People want to feel valued at work. One way to let your team know you appreciate them is to show that you value their time. Through automation, you free team members from the grind of completing mundane tasks over and over, so they can use their skills and abilities on more challenging work.

Embrace the power of automation. With the right tools and some careful configuration, you'll achieve more productivity, greater employee engagement, and higher levels of customer satisfaction. A true win/win/win situation! These Appfire apps can make your work faster and easier.

JSU Automation Suite for Jira

Quick-start your workflow automation in Jira

Perfect for admins who don't want to code

JSU is an intuitive, entirely no-code automation solution. JSU's beginner-friendly UX will help you level up Jira workflows fast.

Learn more and try free

Jira Misc Workflow Extensions (JMWE)

Enhance your workflow automation in Jira

Perfect for admins who need more flexibility without more complexity

JMWE is a flexible Jira workflow automation solution that combines the simplicity of point-and-click with powerful scripting capabilities.

Learn more and try free

Power Scripts

Access advanced levels of automation customization and integration in Jira

Perfect for admins looking for full extensibility, based on simple scripts

Power Scripts is a scalable, scripting-based solution for admins who want to access deep levels of automation, customization, and integration across Jira.

Learn more and try free

HONORABLE MENTIONS

Atlassian Command Line Interface (CLI)
Learn more and try free

Canned Responses
Learn more and try free

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.