Effective Jira Administration

Governance, Workflows, Custom Fields, Best Practices

by Jennifer Choban and Rachel Wright

Produced by

ThinkTilt

Strategy for Jira®
Worksheets, Templates & Materials for Jira

Introduction

Jira began as a tool for software teams, but has evolved into a tool for any team.

Business teams can benefit from Jira's task collection, lifecycle tracking abilities, and built-in reporting. Whether you're on a Marketing team launching your next campaign, a Legal team preparing your next agreement, or a Human Resources team supporting people and processes, you can track your work with Jira.

ThinkTilt and Rachel Wright would like to see more teams experience the benefits of Jira. That's why they formed the partnership **"Teaming Up for Business Teams"** to help you get started, thrive, and conquer your team's "to do" list. We've collaborated on a series of articles to help you on your way. Additionally, some of the worksheets from the **Jira Strategy Admin Workbook** are being added to ThinkTilt's **Process Template library**.

Even if your team isn't technical, you can still use high tech tools. Email, spreadsheets, and shared network drives – be gone! Your team has a real issue tracking database now with Jira.

ABOUT THINKTILT

ThinkTilt was founded to provide powerful support tools that give teams like yours power over your business' processes.

Our premiere product, **ProForma**, is an app for Jira, and is available through the Atlassian Marketplace, both as a fully featured free version, called ProForma Lite, and an unlimited version.

ProForma makes it easy for business teams to build and deploy user-friendly forms with custom forms and fields, backed by Jira's great workflow engine. Empower every team in your organization to take control of their processes and deliver first class request management. All the information you need, where you need it.

ThinkTilt

We challenge you to think of a type of team that wouldn't benefit from Jira!

ABOUT RACHEL WRIGHT

Rachel Wright is an entrepreneur, process engineer, and Atlassian Certified Jira Administrator.

She is the owner and founder of Industry Templates, LLC, which helps companies grow, get organized, and develop their processes.

Rachel also uses Atlassian tools in her personal life for accomplishing goals and tracking tasks. Her first book, the "**Jira Strategy Admin Workbook**", was written in Confluence and progress was tracked in Jira!

Strategy for Jira®
Worksheets, Templates & Materials for Jira

Who’s in charge?

Jira Governance for Business Teams

By Jennifer Choban

Getting off to the right start is always the best way to go. It’s not always reality.

We usually inherit things - business processes, Jira applications, our parents' bad habits, etc. In her excellent resource, the Jira Strategy Admin Workbook, Rachel Wright recommends starting out by creating a Jira Advisory Board. If you're starting from scratch, this is a great first step. If you're already using Jira, now might be the time to put your Board in place, especially if you're considering expanding Jira to other teams in your organization.

Different organizations have different ways of governing their processes. Rachel recommends that the role of the Advisory Board include:

- **Deciding what customizations to create and support in order to strike a balance between giving teams what they need and maintaining a manageable, high-performing application.**
- **Setting standards for privacy, security, and storage and handling of sensitive information.**
- **Developing a process for providing support for teams' Jira projects.**
- **Determining what a successful Jira application looks like. What metrics will define success?**

Who Should be on the Advisory Board?

When you consider how powerful and mission critical Jira can be for your organization, it's clear that it shouldn't be directed by just one person. But who else should be on your Board?

RACHEL RECOMMENDS A GROUP OF ABOUT FIVE PEOPLE INCLUDING:

1. An end user, techy-minded or not
2. A Jira Administrator who understands the application's capabilities
3. A Project Manager, Business Analyst or Strategist – basically, a process-oriented person
4. A high level manager or VP who's ultimately responsible for the work that gets done in Jira
5. A wildcard member to keep everyone on task

Consider having your end user or your wildcard member come from a non-technical business team.

Why Create a Jira Advisory Board Now?

You're probably thinking, we've managed this long without a Board, why do we need one now? If that's the case, one of two things is probably happening. Either your Jira Administrator is handling everything on their own; trying to please everybody or relying on their own knowledge for deciding what should and shouldn't be implemented. Or you do have a group of people who work together to set standards and support Jira users - you just don't think of them as an Advisory Board.

If you don't already have one, the moment of expanding Jira to business teams is an excellent time to establish a Board. Here's why you need one now, even if you didn't think you needed one before:

Expanding Jira to business teams will mean more requests for customizations; more custom fields, more screen schemes, more configurations.

With each request, you will need to decide if it's worth creating and supporting the new asset or whether an existing field, scheme or configuration can be shared. You'll come to better decisions if you include multiple points of view.

You’ll also be collecting more sensitive information.

Consider all the personal information HR keeps on employees. You need a policy to determine what kinds of sensitive information can be stored in Jira. Expanding to business teams also means you’re inheriting all of the privacy and security standards that apply to those teams. Again, you don’t want to be deciding how to navigate that alone.

Teams will have more confidence in Jira.

Finally, teams may be skeptical as to how well a solution developed for IT can address their needs. That’s understandable. We’re all experts in our own areas. Having an Advisory Board that includes non-techies will help people feel more assured that their needs will be considered.

Easily Convert Business Teams to Jira

Expanding Jira to business teams is a great opportunity to bring a tool you already know, love and support to wider use in your organization. Teams from Finance to HR will love handling their requests in Jira, being able to measure and predict their workload using Jira's reporting and knowing that their backsides are covered with Jira's end to end traceability.

Along with making sure business teams' conversion to Jira is done right (the reason you're setting up that Advisory Board), it would also be nice to have it done easily. This is where **ProForma Forms & Fields for Jira** can help. ProForma offers a template library and an easy to use form builder that puts teams in control of collecting exactly the information they require, without the need for custom fields, screens and configurations. You may actually find yourself doing less Jira admin even as you bring more teams into Jira.

You can help your business teams have it all: a great tool, a well-governed application and an easy conversion.

Who Does What in Jira?

System, Application & Project Admins

By Rachel Wright

I was recently asked,"If Jira project admins can now edit their own workflows and screens, what's left for the application admin to do?" Plenty!

Application admins are still very much needed, and their work extends way beyond managing a Jira project. Furthermore, the new project admin abilities aren't as liberating as they may sound. Let's examine the types of admin users.

Types of Jira Admin Users

There are many different types of Jira admin users, and responsibilities vary depending on the type. Admin users generally fall into one of the following categories:

System Level Administrators
Users with the ability to perform absolutely every Jira administration function

Application Level Administrators
Users with permissions to perform most Jira administration functions

Project Owners or Leads
A project's single point of contact, often responsible for project strategy decisions

Project Level Administrators
Users with permissions to manage settings for individual Jira projects (Example: Components, project users, etc.)

While the admin types have distinct abilities, a user can be multiple types of administrators at the same time. For example, an application administrator may also be the owner of a specific Jira project. An application administrator could be a system administrator as well if those roles have been combined. For the differences between application administrator and system administrator permissions, see the "**Managing Global Permissions**" documentation.

Jira Admin Responsibilities and Abilities

Each admin level has a distinct set of responsibilities. Next, we'll address the four admin types as two levels: system/application and project.

System Level Administrators & Application Level Administrators

These administrators need to consider the health of the application, impact to the application, and maintenance implications for each decision and change they make. These admins need to be chosen carefully, audited regularly, and approved by the application owner. Application admins typically have the following responsibilities:

- **Assist the Jira Advisory board in maintaining established standards**
- **Communicate standards, procedures, changes, and maintenance outages to your Jira Ambassador team and end users**
- **Assist end users with user-specific settings**
- **Assist Project Level Administrators with managing settings and maintaining their projects**
- **Complete approved customization requests or suggest alternative solutions within established standards**
- **Manage users, groups, and access**
- **Create and configure new projects, schemes, and assets**
- **Remove projects, schemes, and assets when they are no longer needed**
- **Perform application upgrades**
- **Vet, install, and upgrade apps, plug-ins and integrations**
- **Check logs for, and address errors**

Tip: **Download** this as a worksheet and turn it into a Jira Admin job description!

Project Owners & Project Level Administrators

Each Jira project has a listed "Owner" or "Lead" who is sometimes also the default issue assignee. Additionally, individual projects can have an unlimited number of administrators. As such, there's an opportunity to involve additional users in project-level maintenance and management.

Project admins typically have the following responsibilities:

1. Set and maintain Components, Versions, and other project-specific settings in accordance with established standards
2. Manage users and groups in the "Users and roles" area
3. Routinely triage (or appoint a triage person) to assign and review issues as they are created
4. Maintain the data and accuracy of data in the project space
5. Report any project issues or customization needs to the Jira Support team
6. Respond to questions or approvals requested by the Jira Support team

Additional editing abilities

Additionally, project admins have limited workflow editing abilities in Jira version 7.3 and limited screen editing abilities in version 7.4. Also in 7.4 these abilities can be enabled or disabled through Permission schemes.

Project admins can only utilize assets that already exist. For example, they can add an existing status to their workflow or an existing custom field to a screen, but they cannot remove a status, create or rename statuses, or create new custom fields. They can modify transitions, but not edit transition screens or transition behaviors (properties, conditions, validators, or post functions). Further, these editing abilities only apply to projects where the workflow and the screens are not shared with other projects. If you've been sharing project configurations, as highly recommended in the **Jira Strategy Admin Workbook**, it's possible that few or none of your project admins will have these new editing abilities. Additionally, the default workflow and default system screen still cannot be edited by anyone. Read more about these features in the **7.3** and **7.4** release notes.

How to Check for Workflow Editing Abilities

1. Use the Admin UI

If you have few workflows, you can manually look for ones that are only used by one project. In the Jira Admin UI, visit Admin > Issues > Workflows. Click the “View” link next to each workflow. The following page will show how many projects use the workflow.

2. Use Atlassian’s Script (Jira Server Only)

Atlassian created an **admin helper script** to detect workflows and administrators impacted by the 7.3 change. The script requires node.js and you must be able to execute it on your server.

3. Use the Database (Jira Server Only)

This method is not perfect, but it got me to the data I needed. Work with your database team to improve the sample queries or format them for your database type.

JIRA DATABASE QUERIES

First, I counted the number of projects used by each workflow, looking for any that are not shared

```
SELECT wse.workflow, count(p.pname) AS `Projects Using Workflow` FROM
nodeassociation n INNER JOIN project p ON p.ID = n.source_node_ID INNER JOIN
workflowscheme ws on ws.ID = n.SINK_NODE_ID INNER JOIN workflowschemeentity wse
on wse.scheme = ws.ID WHERE n.source_node_entity = ‘Project’ and n.sink_node_entity =
‘WorkflowScheme’ GROUP BY wse.workflow ORDER BY `Projects Using Workflow`,
workflow;
```

(those with a project count of 1.)

```
SELECT p.id AS project_id, p.pname AS project_name, p.lead AS project_lead, ws.name
AS project_associated_workflow_scheme, wse.workflow AS
workflow_scheme_associated_workflow FROM project p LEFT OUTER JOIN
nodeassociation na ON na.source_node_id = p.id AND na.sink_node_entity =
‘WorkflowScheme’ LEFT OUTER JOIN workflowscheme ws ON ws.id = na.sink_node_id
LEFT OUTER JOIN workflowschemeentity wse ON wse.scheme = ws.id LEFT OUTER
JOIN jiraworkflows jw ON jw.workflowname = wse.workflow WHERE wse.workflow =
‘Workflow Name 1’ OR wse.workflow = ‘Workflow Name 2’
```

Next, I retrieved project details for each of the not shared workflows. I mainly wanted to know the project id, project name, and lead.

I put all the info into a spreadsheet for further analysis. From this abbreviated workflow and project list, I was able to examine individual project settings, like screens and permission schemes, to determine who would be able to take advantage of additional project admin features.

Creating & Managing a Jira Support Project

By Jennifer Choban

The easiest way to manage user requests and support the software is from within Jira itself!

Perhaps it's stating the obvious, but Jira is complex. There are a lot of parts to manage. In order to keep everyone who uses your Jira application happy, you need a way to receive their requests and feedback, to keep track of all of your Jira projects and to manage requests for customizations. In other words, you need Jira to manage Jira.

Creating a Jira Support Project

In the **Jira Strategy Admin Workbook**, Rachel Wright describes how to set up a Jira Support Project which can be used for just this purpose. Your Jira Support Project is the mechanism for:

- Creating and archiving Jira projects
- Managing customizations and requests for customizations
- Responding to user support requests
- Measuring how well you're doing providing support to Jira users

Tip: Follow the **Jira Strategy Admin Workbook** on how to set up a Jira Support Project.

The Workbook provides detailed descriptions of the issue types, fields, screens and workflow you'll want for your Support Project. You can expedite the process of creating a your project by taking advantage of the forms and templates already available in the **ProForma template library**. These include processes for managing requests for:

- **Custom fields**
- **New projects**
- **New user accounts**
- **Permission changes**
- **Project access**
- **Project closure**

Customer Feedback

Great minds think alike. In his presentation **Five Simple Strategies to Jira Service Desk Success**, Greg Warner of **ServiceRocket** recommends creating a feedback request type so that you can constantly be improving.

Rachel takes this a step further, describing how you can ask customers to verify completion of their requests and create a transition screen with a few quick survey questions to collect their feedback. Collecting and using feedback will improve your systems and help users see themselves as valued stakeholders in your Jira application.

The Jira Support Team

A lone Jira Administrator is not a support team. It takes a village. Along with a **Jira Advisory Board**, Rachel recommends having a team of Jira Ambassadors who serve as liaisons between you and the Jira users you support. These could be Project Leads, or other power users who have a good understanding of Jira. Ambassadors are important allies in answering common questions or getting the word out when there's a change to your application or your procedures.

In what is perhaps the most creative suggestion in the **Jira Strategy Admin Workbook**, Rachel suggests creating a few fictional characters to help disseminate information. She uses the Jira Genie and Gerry the Jira Gerbil. Jira users receive a lot of notifications. Fictional personas help messages stand out, get opened, read and shared. (I'd rather not speculate as to why we're more likely to read a message from a fictional character than from one of our actual co-workers...) Plus, it's almost always advantageous to deliver important information with a light and

Limitations

There is one time when Jira is not the best way to manage Jira - during an upgrade or a "lock and rebuild" reindex. Your plans, notes and documentation will be temporarily unavailable at that time. One solution, if you're using **ProForma forms** for your documentation, is to **download a spreadsheet** of the relevant information before starting your upgrade or reindex.

The job of managing your Jira application is a big one. You can be successful with the right tools and the right team.

Four Jira Workflows for Business Teams

By Jennifer Choban

It’s awesome when one tool can be used for multiple purposes. Jira workflows are no exception.

There are a few reasons you should try to limit the number of workflows used within your organization. The first is simplicity. Fewer workflows will make your Jira instance less complicated to manage. A less complicated Jira instance has a good chance of being a better used, better performing Jira instance.

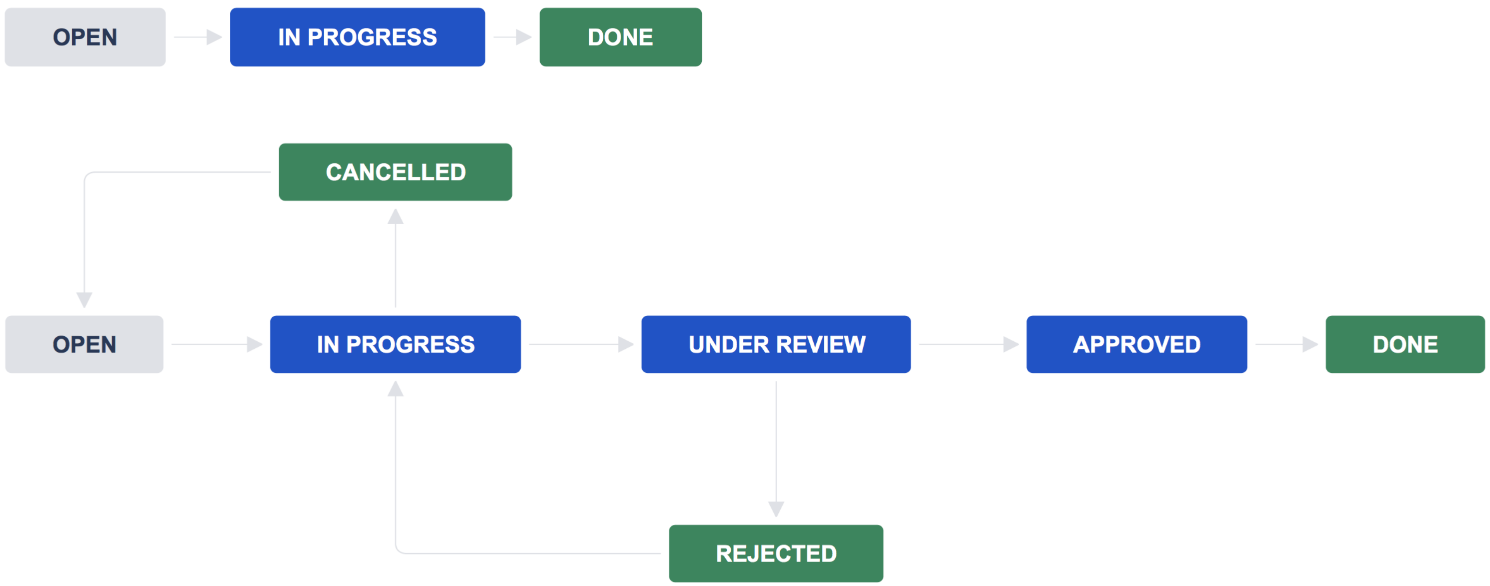
But fewer workflows also makes life easier for your teams' customers. As you start bringing more and more business teams into Jira, you'll notice that a lot of processes involve more than one team. Think about employee onboarding. HR does the heavy lifting by processing reams of paperwork- getting bank details and emergency contact information, ensuring that the new employee has legal working status, setting them up for (sigh) appropriate tax withholdings. But HR is not the only team involved. Your new employee will also need a place to work, a name badge, a key to the building. After Operations has provided those things, they'll need IT to set them up with a computer, an email account, and access to the appropriate networks.

Tip: If you're not going to query for "all issues in a certain status", that status may not be necessary or useful.

Having multiple teams use the same workflow means you can pass requests (move issues) from one team to another. Therefore, rather than the hiring supervisor having to submit three different requests to onboard their new employee, they could submit one.

So let's take a look at a few of the most common workflows and think about how they could function for different teams.

Project Management & Process Management Workflows

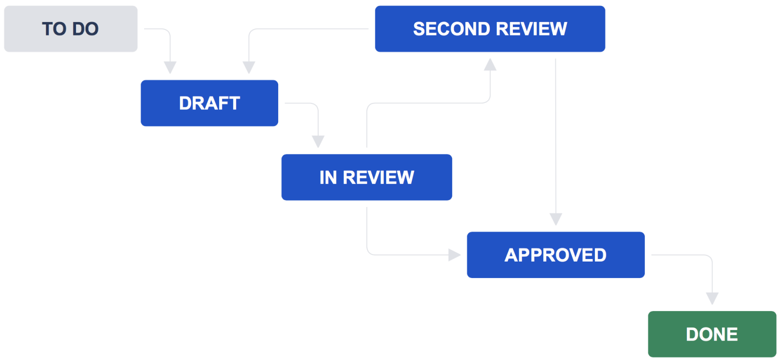


The above are two of the most common workflows. In both cases, work is moved from an “Open” or “To Do” status to a “Done” status. The difference is that the Process Management workflow requires an approval.

A whole lot could get done using just these two workflows. An important thing to remember is that the same status can mean different things to different teams. Take the “Done” status that we use for issues that have been resolved. For a Marketing team creating content “done” may mean published. For a Procurement team “done” may mean purchased. Those differences do not mean they can't use the

Document Approval Workflow

Now let's look at a couple more workflows you might want to have in your collection. Atlassian offers a Document Approval workflow that looks like this:



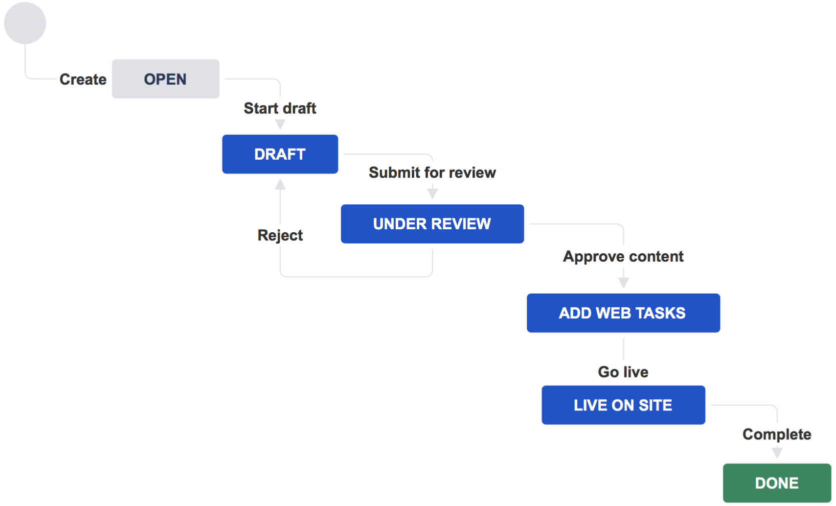
There are two important differences between this workflow and the Process Management workflow:

1. The Document Approval workflow has statuses for two separate reviews before the item in question, in this case
   a document, is approved; and
2. There's no option to cancel - this is a process that has to be completed, the only question is how.

Again, there's no reason this workflow has to be limited to documents. Change "Draft" to "In Progress" and you have a workflow that could work for any number of processes.

Marketing Project Management Workflow

Finally, let's look at this one which was included in an Atlassian blog post entitled **Marketing project management with Jira Core**. This workflow was designed for managing the publication of blog posts:



The interesting thing here that "approval" is not the end of the road. Even after the content has been approved, someone still has to load the article into your blog, set key words, schedule the post to go live, etc.

If the statuses in this workflow were given more generic names, it could be used for other processes - for instance, contract management. Several iterations (from draft to review and back again) may be needed before the contract is actually approved. It needs to be executed (add web tasks) and then needs to be complied with while the contract period is open. Finally, when the contract is complete it can be filed away as "done".

There may be times when these basic workflows won't fit a process, or even times when you need to create a custom workflow, but embracing a strategy of simplicity with regards to workflows will not be limiting, in fact you may find it quite liberating.

Best Practices for Creating a Custom Workflow

By Rachel Wright

If you’re on a software team, you probably use the default Jira workflow or something close to it. But what if you’re on a business team or the default options don’t fit the way you work?

A workflow is a standard set of statuses (steps) and transitions (movement between steps) that each issue follows through in its lifecycle. Statuses take an idea from “conception” to “completion”. Each Jira project can have its own workflow and each issue type within a project can have its own workflow as well. For example, the Legal team has a specific process for contract review and a general process for all “other” types of requests. Their Jira project might include issue types like the standard “Task” and a custom type like “Contract.”

- The “Task” issue type has a very simple workflow, with the statues “To Do” and “Done.”
- The “Contract” issue type requires additional statuses for approval and execution steps that occur in a contract review process.

RECOMMENDATION

In the beginning, keep workflows as simple as possible, until you’ve uncovered a deficiency or process step that needs special attention.

Custom Workflow Tips

1. Before creating a new custom workflow, have the user explain their real life process to you. The workflow should be as simple as possible.

2. First, draw (preferably on paper) a workflow to ensure it makes logical sense and all forward and back transitions are accounted for. You can use the “Custom Workflow Documentation” template in the **Jira Strategy Admin Workbook** or in **ThinkTilt’s Process Template library** as a way to communicate and document workflows.

3. After drawing the workflow, write the workflow out in words. This can uncover additional needs you may have neglected to draw or consider.

4. Include logical backwards transitions so users can self-manage issues.

5. Give users options to abandon or stop progress on issues at appropriate times.

6. Give project-level administrators appropriate options to fix improperly transitioned issues. Example: Include a “reopen” transition button in the final status to address issues that were improperly closed.

7. Use transition conditions sparingly. If a condition is needed, set the restriction to a project role rather than to an individual, for easy maintenance.

8. Use transition validators and post functions to minimize the amount of manual work a user has to do.

   - **Automatically assign an issue to the reporter when moving to an "information needed" or "verification needed" type of status.**
   - **Automatically assign an issue to the Project Lead in a "triage" type of status.**
   - **Automatically move a parent issue to "In Progress" when a child issue starts progress.**

9. Name your statuses

   - **Name statuses so they reflect the current state. Good status names immediately tell a user what is occurring and what state an issue is in the workflow process. For example, "Pending Review", "In Review", "Being Reviewed", "Awaiting Review", etc.**
   - **Make any status names short and indicative of what is happening with an issue. Long, multi-word names are harder to query and may be truncated on certain screens.**

10. Name your transitions

    - **A Transition name should be short and reflect an action taken.**
    - **Good transition names immediately tell a user what action to perform to progress an issue. Example: For an issue in "Pending Review" status, a good transition name would be: "Review Complete." If you need a "pass/fail" situation, where an action must pass a test before a transition can occur, good transition names would be: simply "Pass" and "Fail."**
    - **Bad transition names confuse the user about how to move forward. Example: "Review." A**

Example: Your company is signing a partnership agreement

The contracts process requires a review of the contract by both parties and potential edits before final execution. It’s a predictable process requiring a short workflow like:

Open > In Review > In Execution > Closed

A generically named status like “In Review” is better than a legal-specific name like “In Contract Review”. Other Jira projects can use the generic version regardless of what type of thing needs to be reviewed. You want to share assets and schemes between projects as much as possible.

The Legal team is doing many things in the background that may not need to be reflected in the workflow.

For example, in the “In Review” phase, the Legal team is reviewing the contract, researching legal topics, communicating with internal teams, negotiating terms with the external company, etc.

In the “In Execution” status, the CEO is finding his favorite signing pen, both companies are trading paperwork, and your Legal team is entering the final result into their contracts database.

In the above example, is it useful to create a status for every step that occurs in the contracts process? Do you need to track how many times the contract was modified during the review process? Do you need to track which parties have signed the agreement so far? If the answer is “no” a phased approach may be more useful. It might be more useful to track signature collection in a custom field.

RECOMMENDATION

If you’re not going to report on something (ex: “How many contracts have been signed by us?” in the above example) that status or custom field may not be necessary or useful.

Don’t over-complicate your custom workflow with steps and statuses you don’t really need. Your end users will thank you for it.

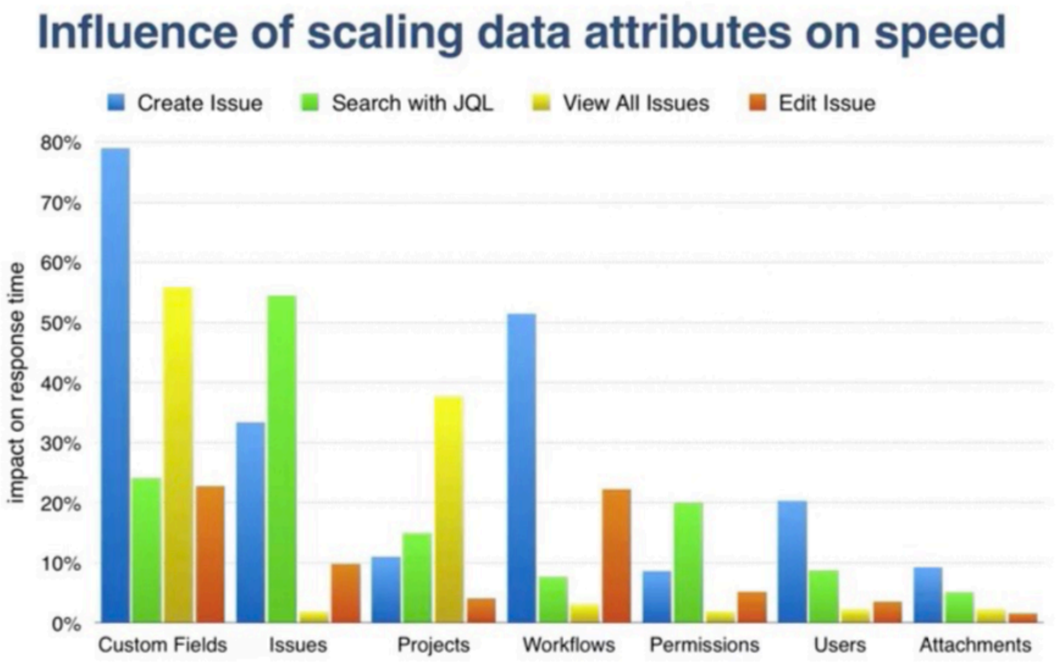
Jira Custom Fields and Their Alternatives

By Jennifer Choban

Jira is a great tool for business teams because it's so easy to customize, but be careful lest you fall into a murky swamp of slow performance and more custom fields than you care to manage.

Customizing Jira for Business Teams

Jira is a great tool for business teams because it’s so easy to customize. After all, the information a software team needs to track will be different from the information an HR team needs to track. You can create custom fields to collect information that doesn’t fit in standard Jira fields, but be careful lest you fall into a murky swamp of slow performance and more custom fields than you care to manage. In fact, Atlassian has identified the number of custom fields as the attribute which has the **highest impact on the speed** of the most common Jira actions.



Wondering if you’ve got too many custom fields already? According to Rachel Wright, author of the **Jira Strategy Admin Workbook** load time of your screens and your custom field admin page are good indicators. Also, note how long your custom field admin page scrolls.

What should you do if you’re committed to limiting the number of custom fields, but still want to take advantage Jira’s flexibility? **Who should decide** when a new custom field is justified? And what should the criteria be for that decision?

Start by putting a process in place that project leads can use for requesting a new custom field. Use the **ProForma process template** or the **Strategy for Jira worksheet** to gather information such as the proposed field’s purpose, type, screen scheme and any needed validation rules.

Then consider the following questions:

- **Will the new field be used by multiple projects?**
- **Will you query issues based on this field?**
- **Will you run a report of all the values in this field?**
- **Will the field duplicate an existing field? (Rachel recommends publishing a list of existing custom fields and their uses to encourage users to make the most of what's already there rather than requesting new fields.)**
- **Will it duplicate core Jira functionality?**

If you determine that a new custom field is indeed needed, pay special attention to creating it with the correct field type, as this cannot be changed. Having the requestor provide examples of the data the field will hold will help ensure that you select the correct field type. (Or maybe you can suggest a better way to track the info altogether!) Give the new field a generic name so it can be used by more than one project.

Alternatives to Jira Custom Fields

Saying no to creating a new custom field, doesn't mean you have to say no to giving users what they need. You have a couple of other options for collecting the same data:

Use a standard field for a custom purpose

Standard Jira fields can be manipulated to collect different data for different projects. For example, the Jira field "Labels" can be used in different ways by different teams. A marketing team using Jira to track their production of marketing assets could use the Labels field to record the marketing campaign an asset is associated with. The same field might be used in a different project by the facilities team to record locations. You can use **field configuration** to set a project-specific description of what should

Adapt an already existing custom field

Field context can be used to make an existing custom field serve different functions in different projects. Field contexts allow you to set a default value and a defined options list for the field within a given project and with a given issue type. For instance, you could create a custom field called "category". The HR team might use this field to store employment status and might have four options (full time, part time, casual, contractor) to select from, with full time set as the default value. The finance team could use the same field in their project, but the options could be set to record different payment methods (EFT, check, purchase order, etc.).

Use a ProForma form

Another option is to collect the information on a form. Forms are an easy way for business teams to collect exactly the information they need without requiring changes and customizations to your Jira configuration. Teams can design, build and deploy forms that gather data structured to their needs and validated with their business rules. Many of the data points required to fulfill a request don't need to be queried or reported on. For those that do, the ProForma form builder makes it easy to "pipe" information from a form field to a Jira field.

Finding the balance that allows you to maximize Jira's flexibility without sacrificing performance is a key consideration when expanding Jira to business teams. **ProForma can help**.

Amp Up the Power of Jira

Four ways of adding forms to Jira

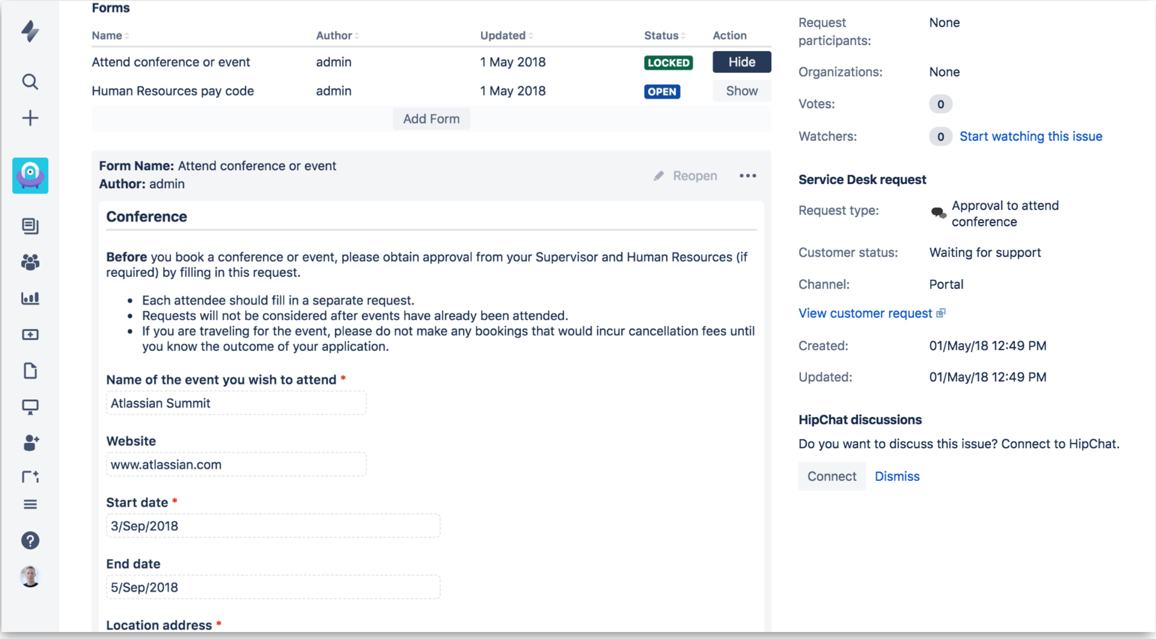
By Jennifer Choban

Wouldn’t it be nice if you could have your cake and eat it too? Embedding a form, or several forms, in a Jira issue is the solution.

Wouldn’t it be nice if you could have your cake and eat it too? Imagine having all of the detailed information you need (structured so it’s easy to find) on a Jira issue without having to create custom fields.

Embedding a form, or several forms, in a Jira issue is the solution. Forms are an enduring business tool because they make it easy to get the precise, detailed data that’s required for a particular process. The information can be structured, formatted and validated to meet your teams’ needs.

While forms can be created for any possible use case (see the ProForma template library for examples), it’s helpful to consider the broad ways that forms can make Jira even more effective.



Forms

Name	Author	Updated	Status	Action
Attend conference or event	admin	1 May 2018	LOCKED	Hide
Human Resources pay code	admin	1 May 2018	OPEN	Show

Form Name: Attend conference or event
Author: admin

Conference

Before you book a conference or event, please obtain approval from your Supervisor and Human Resources (if required) by filling in this request.

- Each attendee should fill in a separate request.
- Requests will not be considered after events have already been attended.
- If you are traveling for the event, please do not make any bookings that would incur cancellation fees until you know the outcome of your application.

Name of the event you wish to attend *
Atlassian Summit

Website
www.atlassian.com

Start date *
3/Sep/2018

End date
5/Sep/2018

Location address *

Request participants:	None
Organizations:	None
Votes:	0
Watchers:	0 Start watching this issue

Service Desk request

Request type:	Approval to attend conference
Customer status:	Waiting for support
Channel:	Portal

View customer request

Created:	01/May/18 12:49 PM
Updated:	01/May/18 12:49 PM

HipChat discussions
Do you want to discuss this issue? Connect to HipChat.

No 1: Background Information

Forms can be used to preserve and display the thinking behind the decision to carry out a particular project. This is useful for Project Initiation Documents, use cases, KPIs and business justifications. Simply put, it allows you to keep the answer to the question, “Why are we doing this?” in front of the team members who are working on a project. This serves to keep everyone motivated and limit scope creep.

No 2: Specifications

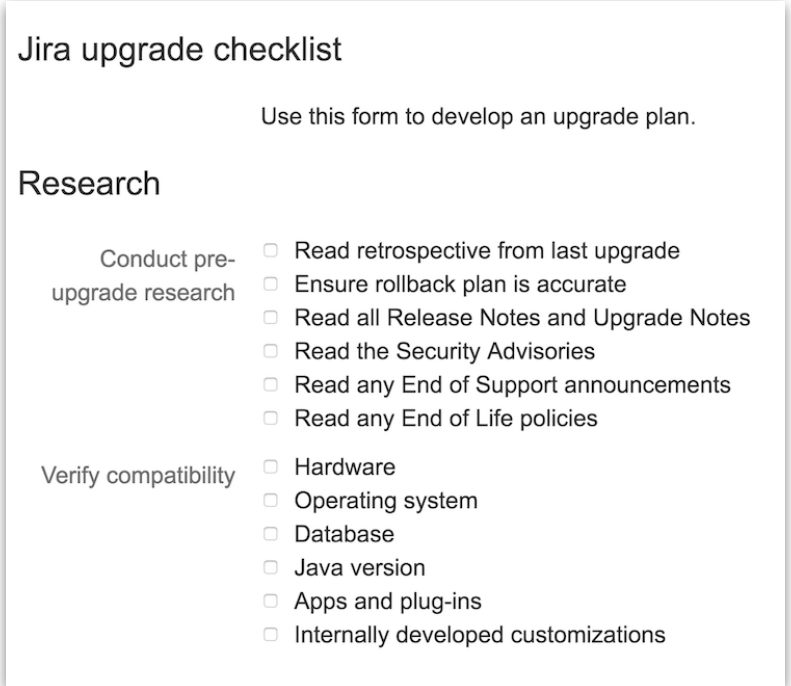
Creating a new product? There are a lot of features that need to be specified. Forms let you break out all of that information into individual fields so it can be easily inputted, viewed and addressed. There's no risk of missing important details that are lumped together in the description field. ProForma includes the option of locking the form, so that the contents cannot be changed once the form has been submitted.

No 3: Request Management

Forms are the right way to get the required information for processing service requests. If customers are not prompted with specific questions, they may forget to include important details when creating a request. This means that the agent will need to go back and ask for the information. The customer gets frustrated by the delay and work in progress piles up for the service agent. Things would have been better for everyone if all of the information had been provided in the first place. In fact, a well-designed request form can be the key to providing one-touch service.

If you're using Jira Service Desk, the ProForma library of process templates makes it easy to bring services into Jira. Each template includes a portal form, request type, workflow and any needed secondary forms, all in one install.

No 4: Checklists



Jira upgrade checklist

Use this form to develop an upgrade plan.

Research

Conduct pre-upgrade research

□ Read retrospective from last upgrade
□ Ensure rollback plan is accurate
□ Read all Release Notes and Upgrade Notes
□ Read the Security Advisories
□ Read any End of Support announcements
□ Read any End of Life policies

Verify compatibility

□ Hardware
□ Operating system
□ Database
□ Java version
□ Apps and plug-ins
□ Internally developed customizations

Forms can also be used to create that mother of all tracking tools – the checklist. Including a checklist on an issue means you can standardize procedures and limit the use of subtasks. Whether the checklist represents acceptance criteria, a definition of done, or an ongoing to do list – you can include as many items as needed and organize them into logical sections for easy navigation. Multiple checklists can be added to a single issue and you can include comment fields for times when checklist items need a little more explanation.

ProForma Custom Forms & Fields for Jira

Fields that are added to forms are fields that don't need to be added to the Jira issues themselves. For items that you need to query or report on, you can link form data to a Jira field with one simple click. This means fewer issue types, fewer custom fields and better Jira performance. Thus, the ProForma app makes Jira more manageable, as well as more useful.

Keeping Track of it All: Jira for Asset Management

By Jennifer Choban

Jira isn't just for software and IT teams. See how teams like Facilities and Operations can use Jira for asset management.

We've talked about what a great tool Jira can be for helping your employees **get the right stuff**, but what about once they have it? You still have to keep track of what's in your inventory, what went to whom, how long said item is under warranty and when it's due for servicing. In other words, you need an asset tracking system.

Beyond Inventory Tracking - Advantages of Using Jira for Asset Management

An effective asset management system will do more than help you track your inventory. It will guide teams throughout your organization in answering vital questions such as:

- **What items need to be included on a depreciation schedule?**
- **How much insurance do we need?**
- **How much money should my department budget for hardware replacements next year?**
- **Was the cost of the extended warranty worth it?**
- **How much time is left on our service contract?**
- **Did that product serve me well? Should I buy it again?**
- **Do I really need 12 more of those or do we have some here that aren't in use?**
- **And of course...where the hell is that thing?**

As a side note, there are numerous stories of individuals using Jira to track their personal assets. Rachel Wright, author of the **Jira Strategy Admin Workbook** describes how she used Jira to track her household items while moving - *and knew exactly where everything was even before getting unpacked!* Others manage their collections – of fine wines, great books, craft beers...

Whether you’re tracking personal or business assets, there are a few things you’ll need, such as a label maker and an asset tracking workflow. In their **eBook on using Jira Service Desk for Asset Management**, Atlassian describes the issue types, custom fields, screens and workflow they use for their asset tracking system. If you’re tracking multiple types of assets - hardware, software, office equipment, furnishings, etc. you will have different properties for each type of item. You can minimize the number of issue types (in this case asset types) and custom fields you need by using a form.

Why Use Forms for Jira?

Along with highly simplified Jira administration, forms provide several additional advantages:

- **Forms are easy. They are familiar. Unless it comes from a tax collecting agency, people generally don't mind filling out forms.**
- **You can put form design in the hands of the teams who do the work. Jira admin - the creation of issue types, custom fields, etc. is best left to technically- minded people who know Jira. However, a straight forward form builder can be used by anyone. You can let the Operations team create the form they want for tracking first aid supplies, furnishings and cans of paint, while simultaneously letting the IT support team create the form they need for tracking monitors and laptops.**
- **In addition to allowing teams to structure their data collection to their needs, online forms make it easy validate for completion, and compliance with business rules.**
- **Form data can be easily aggregated and reported on.**

*There are a many more **templates** for every team, available under a creative commons licence.*

How Do I Get Started?

You can create a thorough and effective asset management system with just three simple tools; a label maker, Jira and ProForma Forms for Jira. In fact, you can even get a head start using the **asset record template** that's included in the **ProForma template library**. The template form can be easily modified by any team to track the assets they are responsible for. Why not add a little order to your life with Jira?

Managing an Upgrade

By Rachel Wright

Take the stress out of upgrades with these the five, high-level steps to ensure a success.

For most Jira Server users, an upgrade is a major activity that requires carefully planning. What is your upgrade plan? How will you prepare? How will you ensure success? How often will you upgrade?

I approach upgrades as five high level steps:

Step 1: Research

Conduct all pre-upgrade "what changed" and compatibility research

This very important first step can determine the success of your upgrade. Start by reviewing the retrospective from the last upgrade so you can improve the upgrade process and plan for issues encountered in the last event. Also, it's a good time to make sure your emergency **rollback plan** is still accurate.

Next, read all of Atlassian's "**Release Notes**" and "Upgrade Notes" for every version between yours and the one you're upgrading too.

After, verify the compatibility of your hardware, operating system, database, java version, add-ons, and any internally developed customizations (Resources: **Jira Requirements**, **Supported Platforms**, and **Checking Add-On Compatibility**).

TIP: Look for changes that might impact the application, users, or user behavior. Look for bugs you've been waiting for fixes for. Also read the **Security Advisories**, **End of Support** announcements, and **End of Life** policy.

TIP: Ask your REST API and database users to read the Atlassian documentation too, so they can prepare for any changes needed in their applications.

Step 2: Pre-Upgrade Tasks (Test Environment)

Copy all production data to lower environments, update plugins, upgrade and test

By now you should know which version you're able to upgrade to. Download the installer file, stop the application, and run the binary. Document the installation process, so you can repeat the steps in production. Review all configuration files, paths, custom files, and settings for accuracy. Also check the logs for major problems.

If all is well on startup, it's time to update the Universal Plugin Manager, other add-ons, and re-index. After the re-index, start your regression testing. Make sure all basic application functions and new features are working as expected.

Finally, contact your REST API and database users so they can verify all is well with their applications. Also, compile any "new features" documentation to share with end users. Conduct an end user and project-level admin demo if UI or feature changes are substantial.

Step 2: Pre-Upgrade Tasks (Test Environment)

Recommendation

Don't have a test environment? Remedy that issue first! Ideally you'll have a secondary server instance but if that's not possible at least create a local instance on your personal computer. Make sure the resources powering your test environment match your production environment as much as possible. Make sure the software version and configuration are an

Recommendation

Before upgrading your test environment, be sure to copy all of your production data to the environment. It's not enough to test an upgrade on a vanilla instance; you need to test it with your specific configuration data!

Mistake!

During testing, I discovered one of my heavily used plugins wasn't compatible with the upgrade version and had moved from free to paid. I clicked the "Buy Now" button on the "Manage add-ons" page, assuming it would take me to a shopping cart with pricing information. Instead, it immediately installed an unlicensed version of the new plugin code! All of our workflows broke and I was inundated by reports of license errors from users. I had to quickly generate a free trial code to restore functionality and sheepishly contact the purchasing department to secure emergency funding for the new plugin. I did all this in production! #facepalm

Step 2: Pre-Upgrade Tasks (Test Environment)

Copy all production data to lower environments, update plugins, upgrade and test

By now you should know which version you're able to upgrade to. Download the installer file, stop the application, and run the binary. Document the installation process, so you can repeat the steps in production. Review all configuration files, paths, custom files, and settings for accuracy. Also check the logs for major problems.

If all is well on startup, it's time to update the Universal Plugin Manager, other add-ons, and re-index. After the re-index, start your regression testing. Make sure all basic application functions and new features are working as expected.

Finally, contact your REST API and database users so they can verify all is well with their applications. Also, compile any "new features" documentation to share with end users. Conduct an end user and project-level admin demo if UI or feature changes are substantial.

Step 3: Upgrade Preparation

Line up support resources, schedule production upgrade activities, and announce plans

At this point, you are confident in the stability of your test environment and ready to schedule the production event. Start by identifying an upgrade team. Who will execute the upgrade? Who will “**smoke test**” the major functions? Who can you contact if there’s emergency?

After you have your team assembled, pick an upgrade time outside of peak use hours. Communicate the upgrade date, time, and expected duration to users and any support teams, like the company help desk or network operations center. Don’t surprise these teams with “*Jira is down!*” reports during the upgrade window!

*Download **sample wording** for your entire upgrade process from the **Strategy for Jira store**.*

Tip: Use Jira’s announcement banner function to communicate upgrade plans.

Sample Code:

```
<div style=”border: 1px solid #9e1c1c; background-color: #fff; padding: 10px;”><b>Upgrade Outage</b><br /
>The upgrade will start on [day], [date] at [time] [timezone] and conclude before the start of business on [day],
[date]. During the upgrade window: (1) you <b>WILL NOT</b> be able to login to JIRA, (2) any changes attempted
<b>WILL NOT</b> be retained, (3) API calls will fail, and (4) issue creation via email will fail. For a list of new
```

Step 4: Upgrade Tasks (Production)

Backup production data, update add-ons, upgrade and test

At last, you've planned as much as possible, know what to expect, and are ready for the upgrade event! It's time to repeat the installation steps you practiced in your test environment including: installation, add-on updates, and regression testing. Use the notes you took in step 2 and be sure to address any differences that exist in the production environment.

Recommendation:

Hopefully you're already taking regular (automated) backups of your database and file system. But when's the last time you verified that your most recent backup occurred and is actually usable? Do that before proceeding.

Step 5: Communication

Announce upgrade and communicate changes and benefits to user base

Finally, it’s time to announce the upgrade to users and complete post-upgrade steps.

Review any previous trouble reports, in case the upgrade remedied them, and be ready to respond to new reports. Check in with your REST API and database users, to make sure all is well with their apps.

Finish any outstanding tasks, compile your retrospective, and make any needed plan updates in preparation for the next upgrade. Also be sure to thank you upgrade team!

Recommendation:

Use Jira’s announcement banner function to communicate the upgrade is complete. Include a link to the “**new features**” documentation you compiled in step 2.

Detailed Upgrade Plan

A well-crafted plan can help ensure upgrade success. Download the **sample upgrade plan** worksheet. Customize it to fit your needs and environment. This worksheet may contain more or fewer steps than necessary for your situation, but it gives you a great starting point. Don’t forget to update and improve the plan after each upgrade.

Vetting Jira Apps and Plugins

By Jennifer Choban

In a Marketplace that offers literally thousands of apps and plugins, how do you know which ones will be truly beneficial? Create a solid process for nominating, vetting and installing Jira apps.

Do you get sensory overload when you visit the Atlassian Marketplace? The current count shows over 1,600 offerings for Jira alone. Too many choices can be paralyzing. How can you identify which ones will be truly useful and worth the effort of maintaining?

A Process for Requesting Jira Apps & Plugins

Start by setting up a process by which teams can request an app. At very least, teams need to provide the name of the app and the url in the Marketplace.

Getting more information, like what the app does, why the team thinks they need it and examples of how they will use it – will give you a solid basis for making your decision.

After you've had a chance to check the price and ensure that the app is compatible with the your Jira application, you'll also want to confirm that leadership has approved the app and is willing to pay for it.

What to Ask When Vetting Jira Apps & Plugins

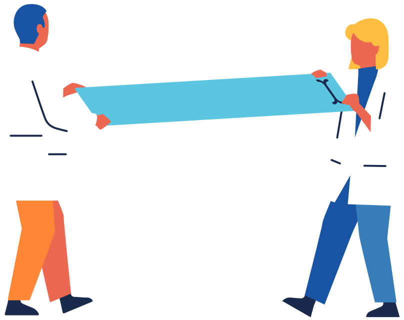
Questions to consider when deciding whether or not to install a Jira app might include:

- **Does the app/plugin duplicate functionality that is already available, but underutilized in Jira?**
- **How much will the app/plugin be used? How frequently? By how many users? How many teams? In how many projects?**
- **Does the app/plugin allow you to leverage Jira, making Jira relevant and useful for more teams?**
- **Does it come from a trusted source?**
- **Is the app/plugin a good value compared to other solutions for resolving the problem?**
- **Does the app/plugin provider offer adequate support?**

Teams requesting new apps are doing so because they believe the app will resolve a specific problem or pain point. New apps, however, also impact the Jira Administrator and Jira Support Team. Every app will add another layer to manage. However, some apps may actually relieve some of the Jira Administrators pain points as well (too many custom fields, too many custom workflows, etc.). An app is of greater value if it's useful for both teams (and different types of teams) and Jira support staff.

Best Practices for Installing Apps

You will of course want to test the new app before deploying it into production. In her excellent resource, the **Jira Strategy Admin Workbook**, Rachel Wright outlines a clear procedure for testing new apps. She also provides a provides a **packet of worksheets** which can be used for vetting, installing and announcing new Jira apps.



Jira Clean-up Time

By Jennifer Choban

No matter how well you govern, stuff accumulates. Learn how to audit, archive and groom your Jira instance.

Depending on where you are in the world, you may be smack in the middle of spring – as in time for Spring Cleaning. Even if it isn't spring, it's a good idea to occasionally audit your Jira instance, archive elements that aren't being used and revisit your configuration to ensure that it's optimized.

Audit

The first step is taking stock of what you have. To do this, start with the "System info" page and note the number of projects, issues, custom fields, workflows, etc. Next, visit the Add-ons admin page and the admin area for each scheme and project asset.

Rachel Wright offers a **worksheet** for recording this information. Alternatively, you could use a spreadsheet or create a form in ProForma. The important thing is that you will want to track the number of Jira assets over time.

As you go through the admin pages, note down assets that are unused (such as a field that doesn't appear on any screens), duplicated (or perhaps similar enough in purpose that only one field is needed) or inactive.

Once you have a complete picture of your current application, you can set goals for your clean-up. Along with removing assets that aren't being used, goals could include reducing the number of **workflows** for easier support and reducing the number of **custom fields** for better performance.

Archive

Before you begin pruning out unnecessary elements, backup and verify your data. Also, make sure you have a rollback plan in case any of your changes cause unanticipated problems.

Elements

Go to the admin page for each element type and use the "identification" column to weed out unused items. The **Jira Strategy Admin Workbook** includes detailed instructions for easily identifying unused elements. In some cases (for example, if you are eliminating an issue type), you may need to migrate issues before deleting.

Deleting unused items is the first step. Depending on your goals, you may also want to consolidate custom fields or workflows. When consolidating fields, consider which field is more widely used, which has a better name, and which can be more easily deleted. Again, you may need to migrate data before you can delete.

Projects

Along with pruning down Jira elements, you should also check for stagnant projects. Use the **Jira Strategy Admin worksheet** or the **ProForma form** to identify projects that have an inactive project lead, few issues or no recently created issues.

You have several options for dispatching completed or stagnant projects:

- **Prevent new issues from being created**
- **Mark the projects as read only**
- **Hide the project**
- **Archive the project**
- **Export the project**

Rachel Wright describes the appropriate use, as well as the implementation steps, for each method in the **Jira Strategy Admin Workbook**.

Users

Finally, you need to address users who have left the organization. You will not be deleting these users because you want to retain the history of their actions. Rather, you will be setting departed users to "inactive." However, before doing so, Rachel recommends the following steps:

- **Move any not closed, assigned issues to the user's supervisor.**
- **Move any not closed, reported issues to the user's supervisor.**
- **Remove unshared custom dashboards, filters, filter subscriptions and boards.**
- **Remove favorite designations for dashboards.**
- **If a dashboard is used by others, move the dashboard to the supervisor or a generic user account. Remove favorite designations for filters.**
- **Move shared filters to the supervisor or a generic user account.**
- **Reassign project leads to the supervisor.**
- **Reassign component leads to the supervisor.**
- **Remove filter subscriptions.**
- **Remove draft workflows.**
- **Reassign agile boards to the supervisor.**
- **Check workflows for any auto assignment transition behaviors.**
- **Make the user account inactive.**

Making the time to regularly review and tidy up your application will make your ongoing admin duties easier and will keep your application clean, relevant and high-performing.

Thank you!
We trust that you found this book helpful.

For more resources and information, please visit:

www.thinktilt.com

or

www.jirastrategy.com

Effective Jira Administration

CONTAINED IN THIS BOOK

1. Establish a Jira Advisory Board
2. Be Clear on Who Does What: System, Application & Project Admins
3. Create a Jira Support Project to manage Jira.
4. Use Common Workflows to Support Multiple Teams
5. Learn the Best Practices for Creating a Custom Workflow
6. Learn about Jira Custom Fields and Their Alternatives
7. Use Forms to Amp-up Jira
8. Jira for Asset Management
9. Managing a Jira upgrade
10. How to Select Jira Apps
11. Jira Clean-up Time

Download other resources and ebooks at