Atlassian Confluence

# Software Team's Guide to Confluence

# About the Authors


### Terrence Caldwell

Product Marketing Manager,
Atlassian


Terrence Caldwell has been on the Confluence marketing team for 3+ years. He's passionate about helping customers get the most out of their experience in Confluence. He spends his time creating content and working with customers and with the product team to optimize that experience. Outside of work you can find him obsessing over, playing, and attending sporting events.


### Jamey Austin

Content Marketing Manager,
Atlassian


Jamey believes Thomas Mann had it right when he said, "A writer is someone for whom writing is more difficult than it is for other people." When he isn't twisting his beard staring at words on a screen, Jamey might be wandering the streets listening to Spotify, teaching his kids the finer points of rochambo, indulging a basketball jones, or generally spending an inordinate amount of time feeling completely amazed by it all. Find him on Twitter: @jameyaustin.

# About the Authors

### Ryan Anderson

Product Marketing Manager,
Atlassian

Ryan is a Product Marketing Manager for Confluence at Atlassian. He has worked closely with the development and product management teams to ship great software to customers for over five years.

### Sarah Goff-Dupont

Editor-in-Chief,
Atlassian

Sarah has been involved in software for over ten years: testing it, automating it, and now writing about it. When not at work, she can be found reading contemporary fiction, smashing it out and keeping it real at CrossFit, or rolling around on the floor with her kids. Find her on Twitter: @DevToolSuperfan.

### John Wetenhall

Product Marketing Manager,
Atlassian

John is the resident product marketer for Confluence at Atlassian's Sydney office. John is passionate about getting more done with Agile (yes, even in marketing) and helping people get the most out of Confluence. When not at the office, John is an avid runner, sailor and music fan. Get in touch on Twitter: @jwetenhall.

# Contents

# More than just issues and code

## By Terrence Caldwell

Software projects require much more than just issues and code.

A lot goes into shipping great software. Hours spent coming up with ideas, translating ideas into issues, writing lines of code to bring those ideas to life, testing, releasing, and continuing to improve on that release with bug fixes and enhancements. But even all of that is only half the story.

A ton happens behind the scenes: customer interviews, feature requirements, release plans, team retrospectives . . . you know the drill. Your team is constantly working to add detail, context, and even delight to your product. That's where Confluence comes in.

Confluence is the home for all the information your development team needs to keep projects moving forward. It's where you assemble the plans, requirements, decisions, and documentation behind your software. It's also the place that connects your development team to product managers, designers, marketers, and other colleagues who participate in the software development process. *Confluence is your common ground*. It's the one place where anyone can access and contribute to what you're working on.

This guide is a collection of pro tips and step-by-step tutorials—tailor-made for software teams—that'll show you how to use Confluence for creating, organizing, and documenting the information your team needs to make awesome products.

We hope you enjoy using it as much as we enjoyed creating it. Cheers!

Chapter 1

# How to create insightful customer interview pages

**By Jamey Austin**

Using Confluence to document and share customer interviews can help with every step of the process, including the most important, final step: using customer feedback to create brilliant products and features your customers will love.

If you're currently putting your customer interview data in a Word doc, Evernote file, or ______ (fill in the blank) and sharing with your team via email, you're in for a treat.

## How to standardize your customer interviews by creating a page template

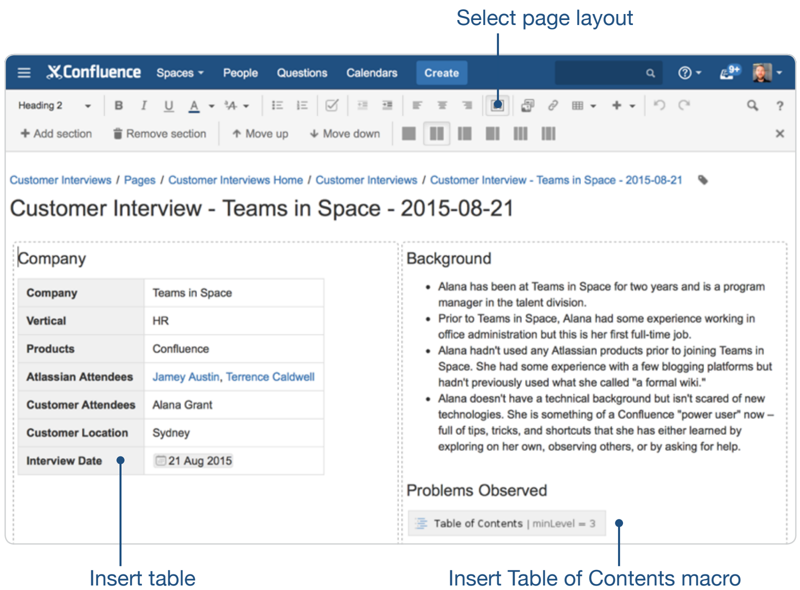
We have loads (technical term) of customer interviews documented on our internal Confluence instance, so we thought we'd share our best practices for creating an interview page template. Feel free to make modifications to better suit your process. Remember, Confluence is a canvas to design as you like.

### Create your first customer interview page

Start off with the essentials. Design your page so you can add company and background information where it's easily accessible. You're providing context for your team, so include details about your relationship with the interviewee, the role of that person, and how the interview came to be. Also, make sure to add what product(s) they use.

The goal is to capture the critical details at the top of your page so anyone can quickly grasp who you interviewed, why, and what you talked about.

A design for your page might look something like this:

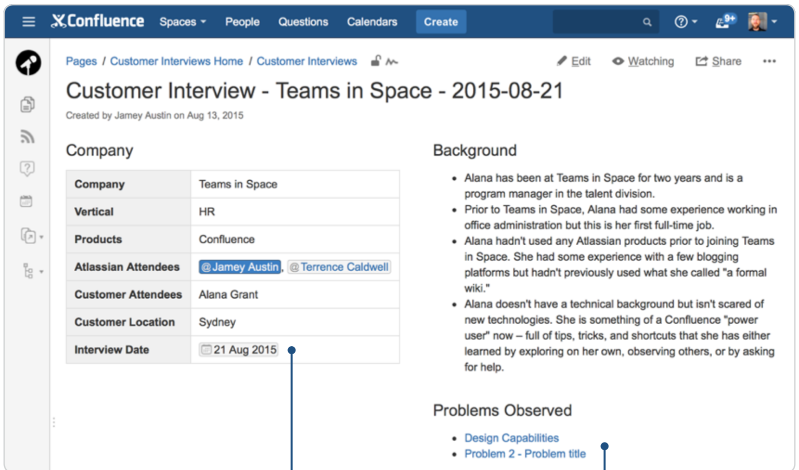



The top section of this page is set up to capture the essential interview details we've talked about. It features a detailed title, and company and background sections.

To create a page with a similar design, try this:

1. Use the **Page layout** button from the editor toolbar (see screenshot) to decide how you'd like your information displayed. Here a **Two column section** is selected which splits the page into two equal sections.
2. Add page elements. Select **Insert table** from the toolbar to add a table like in the above company section and add as many columns and rows as you need.
3. We've inserted a **Table of Contents macro** (select **+** button in toolbar > **Other macros** > **Table of Contents macro**). This macro creates a list of heading links for quick access to more detailed information. In this example, a list of headings for each problem observed.

4. This is what it looks like after you save the page:




Keystroke tip: type // and select from date-picker

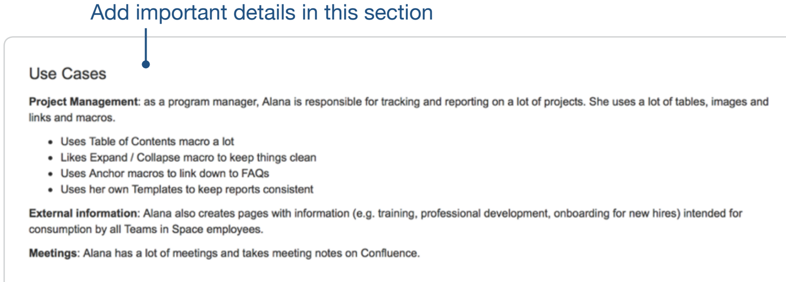
List of headings created by Table of Contents macro

Note the list of headings created by the macro. The information these heading links are pointing to is described in the next section.

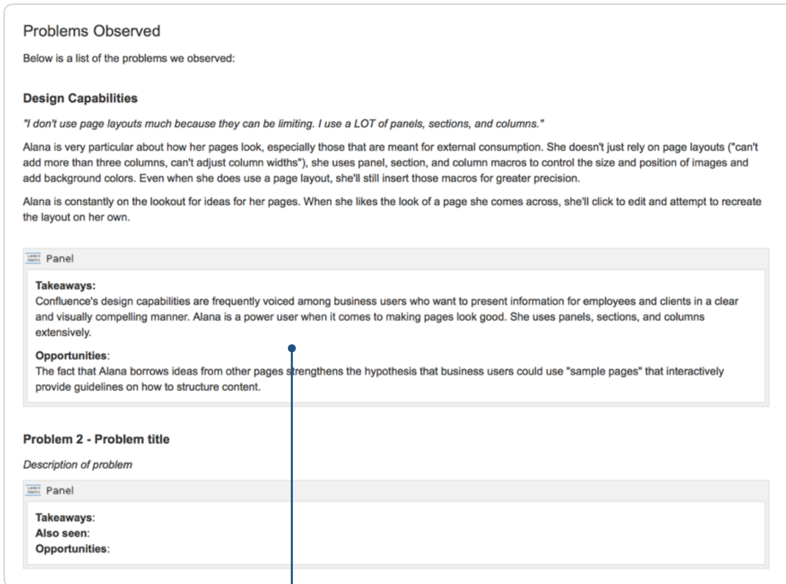
> **Pro tip:** The Table of Contents macro creates a list of headings (in this case, each problem title) that link to a specified section where the full details live. In this example, the headings appear where we inserted the macro in the top "Problems Observed" section and link to the detailed information about each problem in the next section. You can configure your Table of Contents macro to grab headings and create similar lists by selecting edit in the macro and completing the necessary fields. Learn more about the Table of Contents macro.

Now, create a section for the details of the problems collected from the customer. This is how you begin to turn interviews into insights. In our example, there's a place to add use case details:

Add important details in this section



and an area set up to capture the minute details of each problem:




Insert Panel macro

Note the Table of Contents macro used in the first section grabbed the headings "Design Capabilities" and "Problem 2—Problem title," which link users here where the full details of each problem are found.

To design the next section of your page like the above:

1. Create a new section (select **+ Add section** from toolbar, then choose **Single column section**). Our example starts of with use cases.
2. Below that, in the same section, is "Problems observed." To organize this information, we've inserted a Panel macro. (Select **+** from the toolbar > **Other macros** > **Panel macro**.)

**Pro tip:** For each problem, try this method:

1. Write a one-line problem statement. Be concise and articulate the user's intention but not their feature request.
2. Note all observations that led you to confirm the problem. Be sure to include screenshots, recordings, or practical examples to provide additional context for your team.
3. Add opportunities you have to solve the problem. Link to related epics, tasks, or feature requests in JIRA (or whatever you use to manage your team's backlog).

> **Quick tip:** Add **images** to your customer interview pages. Company photos and product screenshots make your customer interview pages better, and adding them in Confluence is drag-and-drop easy. (Or select **Insert files and images** from the toolbar. Also easy!)

### Create a customer interview page template

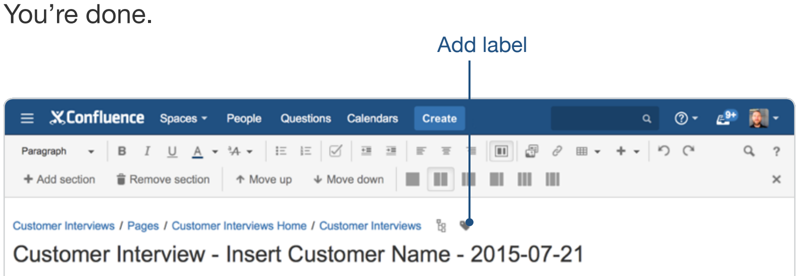
Once you've created a customer interview page you like, Confluence lets you "templatize" that page so you can use it again and again. Learn more about creating a template.

By standardizing your customer interviews with a page template, you'll establish consistency for everyone who conducts customer interviews. The next steps involve organizing, centralizing, and sharing your customer interview information.

## How to organize your customer interviews

### Add labels to your customer interview page template

Labels make searching for customer interviews simple. At the top of the page where your title ends, select the label icon. Add a label like "**customer-interview**." Boom! You're done.




Choosing a standard label for every customer interview helps with organizing all your interviews.

### Add additional labels

After you've added the "**customer-interview**" label to your page, you can add additional **topic labels** (product, use case, terminology) at any time. Labels are important in Confluence because they can be pulled into summary lists on an index page via macros. It's a good idea to get in the habit of tagging up your interview pages using labels. This will give you several display options for your index page, like a list of all customer interviews or a list of popular topics.

## How to centralize your customer interviews

Another great aspect of Confluence is not only keeping your information organized, but also centralized. Here are some suggested steps for centralizing your customer interviews in Confluence.

### Create a customer interviews index page

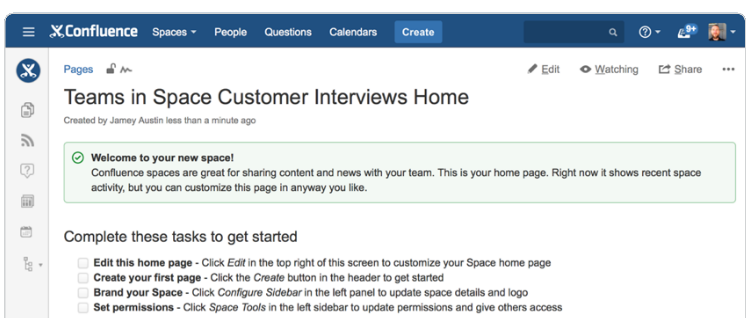
It's a good idea to create a customer interviews **space** and make the home page a customer interviews index page. An index page is great for keeping all your customer interview information in one place.

If your team is small, you may not need a dedicated space for customer interviews. But if you have multiple teams, it makes sense to create a customer interviews space and make the home page your customer interviews index page. Either way, an index page is great for keeping all your customer interview information in one place.

### Tips for creating a space and an index page

*Create your customer interview space*

- Select **Spaces** from the header > **Create space** > **Blank space**
- Add Space name. Add a Space key, or Confluence will create one for you. After you select **Create**, your space home page looks like this:

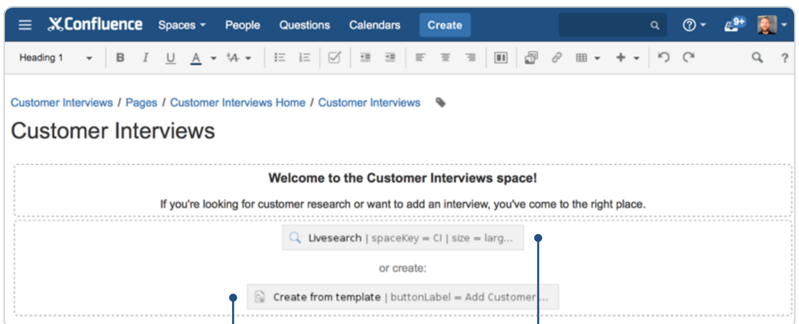



Important note: this is the home page. Complete the first task by creating an index page (details below), and the index page becomes the home page.

## Create your customer interview index page

- Select **Edit** from your customer interviews space home page (or from any page if not using a dedicated space)
- Start creating your customer interviews index page!

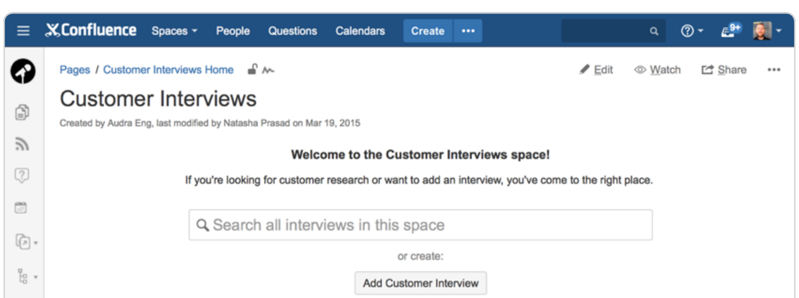
Check out this example from our own Atlassian customer interviews index page, in edit mode:




Use the Create from template macro for the customer interview page you designed

Add the Livesearch macro to find info quickly

And here's what it looks like when you save:

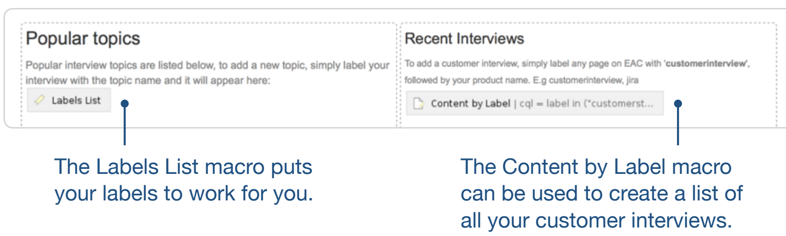



As you can see, the most important information is front and center. You've come to this page for two reasons: you're looking for a customer interview, or you're interested in creating a new one. These two options are at the top of the page, above the fold.

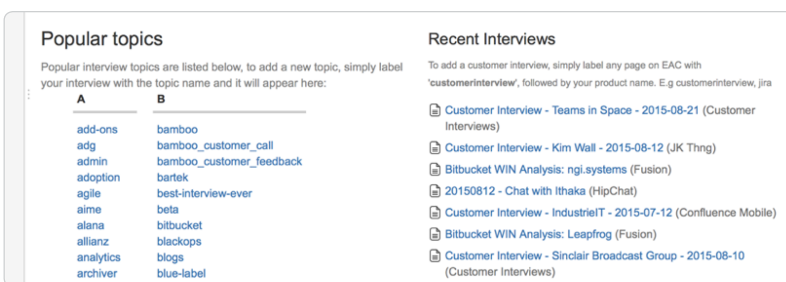
**Pro tip:** Two macros are doing the heavy lifting in this example. Let's take a quick look at each:

1. **Livesearch macro**: lets you add a search box, which can be configured to best suit your needs for your index page. Find out more about the Livesearch macro.
2. **Create from Template macro**: add a button on your index page that opens a page template (like your newly created customer interview page template) when clicked. Check out more about the Create from Template macro.

Your index page can also contain lots of other information that will be helpful to your users, and macros (yes, more macros!) can help you display it. Check out this example, in edit mode:




And here's what it looks like after you save:



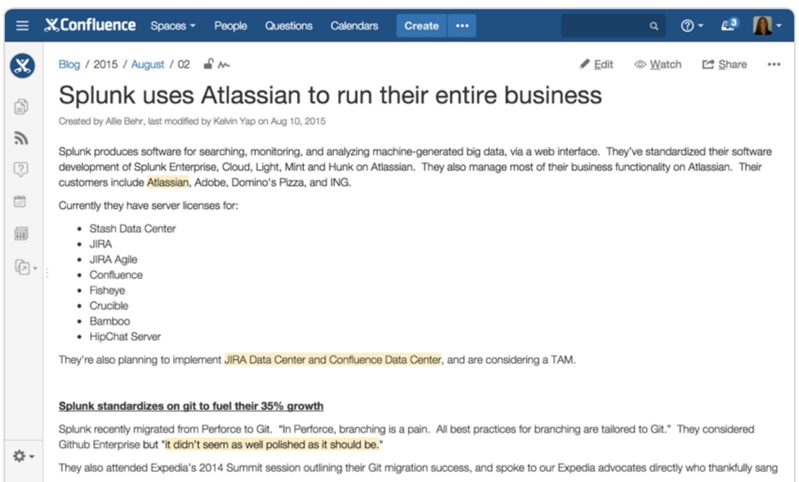

You can organize the information you need on your index page however you wish, but these design ideas should get you going in the right direction. Just remember, you have control of the parameters of these macros and you can create good looking pages and powerful reports. Happy creating!

## How to share your customer interviews

The final step of your customer interview process is sharing the information. You can do this in two simple ways:

1. Use the share button and send a customer interview to anyone on your team or in your organization
2. Blog your product's successes and "room for improvement"s with your organization

Example shared blog post: "Splunk uses Atlassian to run their entire business"




## At a glance: what did you just learn?

Creating a customer interview page—and an index page to organize and centralize all your customer interviews—will help you to both communicate customer feedback to your team and organization and turn interview information into insights about your products. Remember to experiment with you pages to make them work best for your process.

Here's a list of what we covered:

- **Page layouts:** organize your page with different column layouts.
- **Table of Contents macro:** create a list of heading links for quick navigation.
- **Panel macro:** helps organize information by creating a container.
- **Creating a template:** pages used again and again can be turned into templates.

- **Adding labels:** labeling pages does two things: organize information and help macros do their magic.
- **Creating a space:** a dedicated customer interview space is a good idea for any team.
- **Creating an index page:** make the home page of your space an index page for quick reference.
- **Livesearch macro:** add search to help quickly locate customer interviews.
- **Create from Template macro:** add a button to open a new page with the template you created from your page.
- **Labels List macro:** create a quick reference list from the labels you attach to pages.
- **Content by Label macro:** use this to create a list of all customer interviews.

Chapter 2

# How to write product requirements

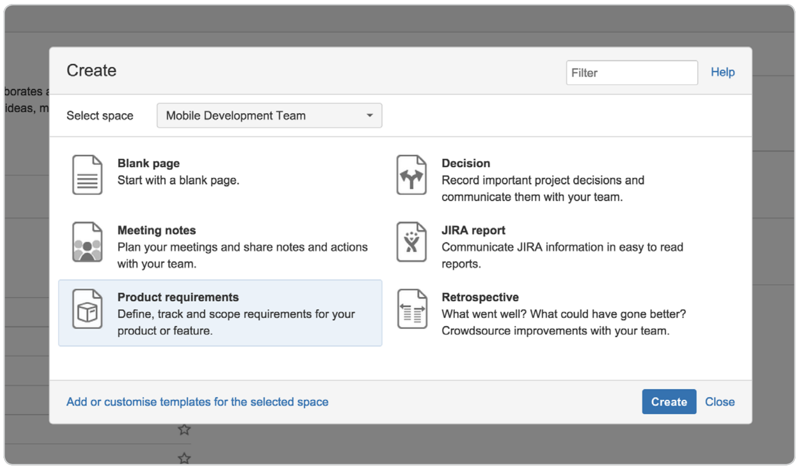
**By Terrence Caldwell**

## Creating a collaborative product requirement

We've written quite a bit about the conundrum of writing requirements in an agile environment. The most important takeaway is: the best way to kick-off your software projects is to build a shared understanding amongst your team. At Atlassian we find the best way to do this is to create a collaborative product requirement to hash out details for large complex epics. When your requirement doc is created and the details are all in one place, it makes it a lot easier for Development and Design to provide immediate feedback and input. When this happens you can iterate quickly and implement work right away. Here's how you can create that requirement and build a shared understanding in Confluence:

### 1. Start with our Product Requirements blueprint

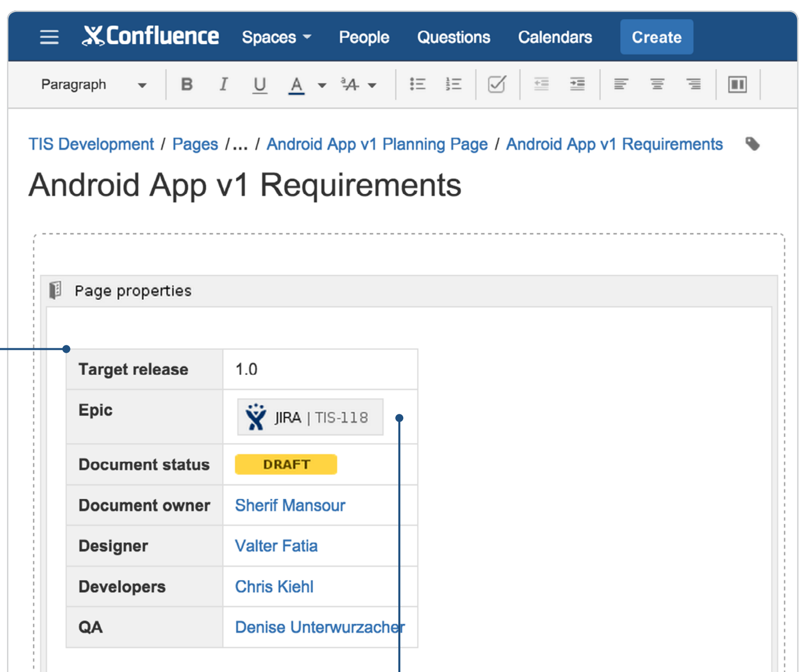
Get started with the special template that we've included. The template will help point you in the right direction, and provide some best practices for what content to include.




## 2. Communicate key details

Your requirements page might be laser focused on a particular feature, or it might include requirements for a release that includes multiple features. Either way, you should capture key high level details at the top of the page for easy reference.

Use a table to define project specifics. Use @mentions for participants and the 'Status' macro for document status.




We add a row to define the related JIRA epic. Just copy and paste the JIRA url to embed it in the table.

- Participants: Who is involved? Use *@mentions* to include the product owner, and other stakeholders.
- Status: What's the current state of the program? Use the status macro to quickly show if you're on target, at risk, delayed, deferred, etc.
- Target release: When is it projected to ship? If you use JIRA, link to your JIRA project.

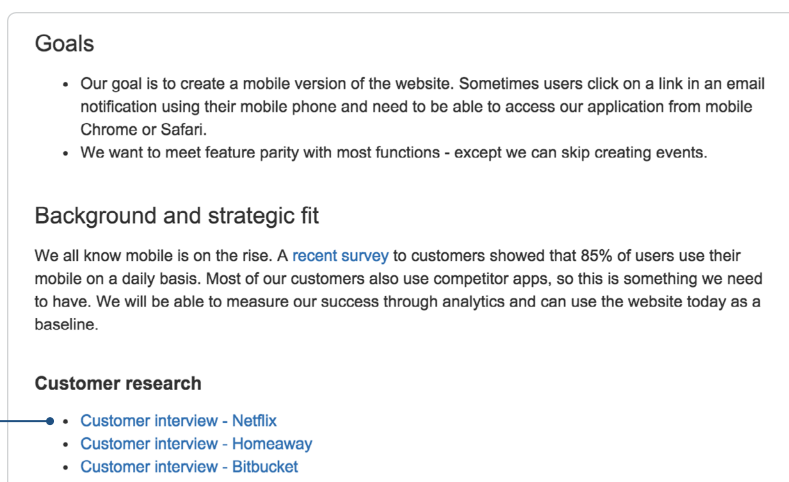
> **Pro tip:** When you create your first product requirement doc in Confluence you'll notice the project details are in a table located in a *page properties* panel. The page properties macro is a powerful macro that allows you tocreate a summary page that pulls in information from multiple pages. In this case, a summary page could bring in each of the fields in your table so you can see who is involved or the status of a given requirements page. Keep reading to see what that summary page looks like.

## 3. Outline goals, business objectives, and strategic fit

The team goals and business objectives section is meant to give you a chance to add some color and context to your requirement.

A good way to think about your goals is from your customers perspective. What exactly are they trying to achieve? How does the solution you're proposing fit into it?

This is a great place to link to other Confluence pages to provide more detail including customer interviews and planning pages.


Goals

- Our goal is to create a mobile version of the website. Sometimes users click on a link in an email notification using their mobile phone and need to be able to access our application from mobile Chrome or Safari.
- We want to meet feature parity with most functions - except we can skip creating events.

Background and strategic fit

We all know mobile is on the rise. A recent survey to customers showed that 85% of users use their mobile on a daily basis. Most of our customers also use competitor apps, so this is something we need to have. We will be able to measure our success through analytics and can use the website today as a baseline.

**Customer research**

- Customer interview - Netflix
- Customer interview - Homeaway
- Customer interview - Bitbucket

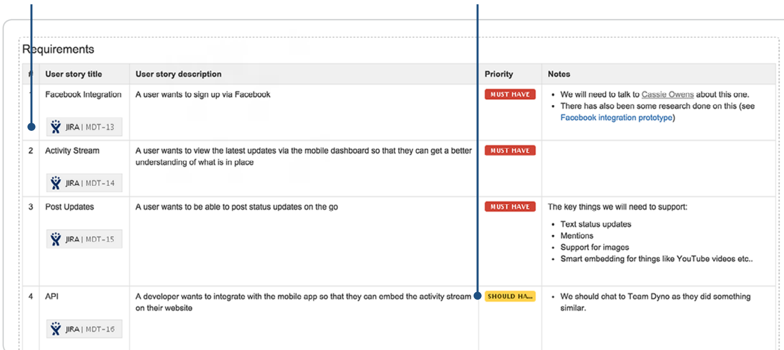
Link to relevant Confluence pages for additional info like detailed customer interview write-ups.

## 4. Create user stories

Use a table to list out your user stories, along with description, priority, and notes. This table is another great place to link to more detailed research pages, and customer interviews.

You can convert your user stories to JIRA issues in Confluence.

Use the Status macro to identify the priority of each story.


Requirements

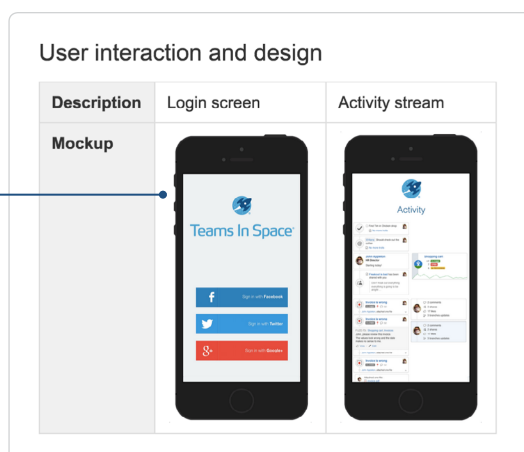
| # | User story title | User story description | Priority | Notes |
|---|---|---|---|---|
| 1 | Facebook Integration<br>JIRA \| MDT-13 | A user wants to sign up via Facebook | MUST HAVE | • We will need to talk to Cassie Owens about this one.<br>• There has also been some research done on this (see Facebook integration prototype) |
| 2 | Activity Stream<br>JIRA \| MDT-14 | A user wants to view the latest updates via the mobile dashboard so that they can get a better understanding of what is in place | MUST HAVE | |
| 3 | Post Updates<br>JIRA \| MDT-15 | A user wants to be able to post status updates on the go | MUST HAVE | The key things we will need to support:<br>• Text status updates<br>• Mentions<br>• Support for images<br>• Smart embedding for things like YouTube videos etc.. |
| 4 | API<br>JIRA \| MDT-16 | A developer wants to integrate with the mobile app so that they can embed the activity stream on their website | SHOULD HA... | • We should chat to Team Dyno as they did something similar. |

**Pro tip:** It's really easy to turn the user stories you write in Confluence into JIRA issues. When the page is saved, just highlight the user story text, and Confluence will give you the option to create JIRA issues. If you have a full table full of user stories, you can convert them all into JIRA issues at once.

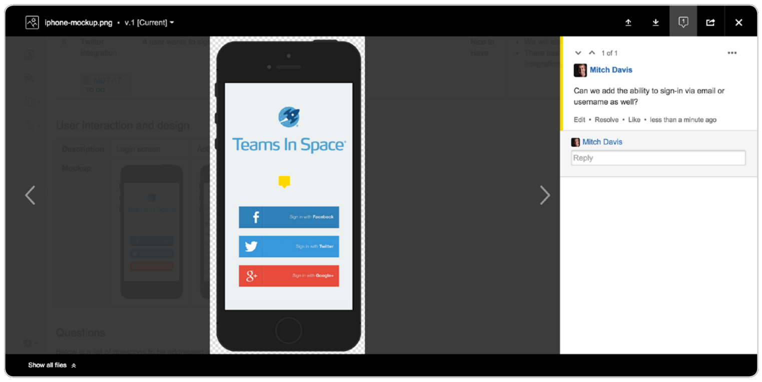
## 5. Review user experience and design

Drag-and-drop design files to embed them on your page. If you're embedding multiple design assets use the attachments macro to create a quick repository of all the files that are attached to the page.

Embed all of your designs, mockups, wireframes, and prototypes. Just drag and drop them on the page.



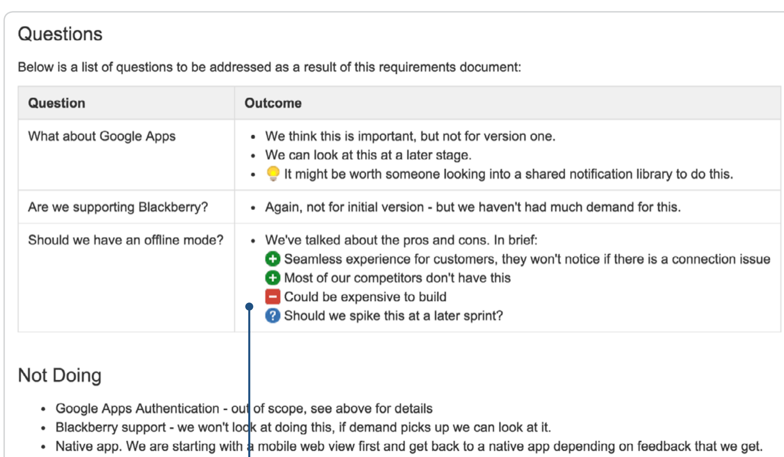
**Pro tip:** Confluence is perfect for collaborating on wireframes, mockups, and design prototypes. Use the rich file viewer to prievew PDFs, images, and powerpoint presentations. You can pin comments directly on files to provide feedback in context. Your design, and dev team can spar on early mockups on the same page.




## 6. Address questions and clarifications

As the team understands the problems to solve, they often have questions. Create a table of “things we need to decide or research” to track these items.

Keep the team focused on the work at hand by clearly calling out what you’re not doing. Flag things that are out of scope at the moment, but might be looked at later.



Questions

Below is a list of questions to be addressed as a result of this requirements document:

| Question | Outcome |
|---|---|
| What about Google Apps | • We think this is important, but not for version one.<br>• We can look at this at a later stage.<br>• It might be worth someone looking into a shared notification library to do this. |
| Are we supporting Blackberry? | • Again, not for initial version - but we haven't had much demand for this. |
| Should we have an offline mode? | • We've talked about the pros and cons. In brief:<br>Seamless experience for customers, they won't notice if there is a connection issue<br>Most of our competitors don't have this<br>Could be expensive to build<br>Should we spike this at a later sprint? |

Not Doing

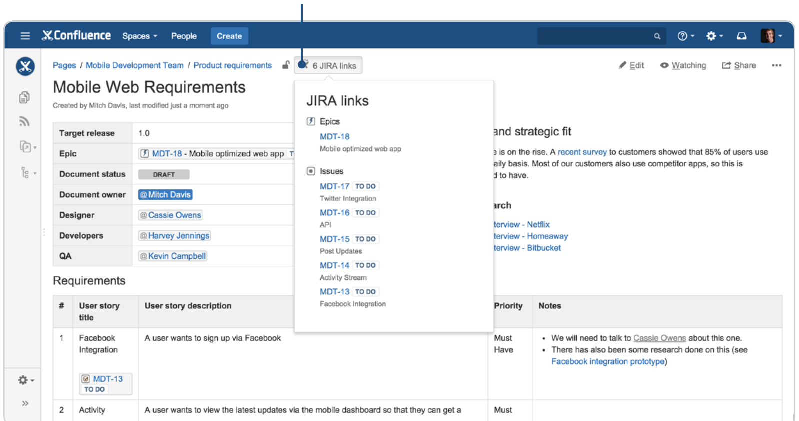
- Google Apps Authentication - out of scope, see above for details
- Blackberry support - we won't look at doing this, if demand picks up we can look at it.
- Native app. We are starting with a mobile web view first and get back to a native app depending on feedback that we get.

Start discussion with open questions. Use emoticons to highlight pros and cons.

## 7. Trace requirements to JIRA issues

Most likely you and/or your development team spend a lot of time in JIRA. It's really easy to get from your Confluence requirement to your JIRA epic and vice versa. You can view all the JIRA links on any Confluence page, by clicking on the *JIRA links button* at the top of your saved page.

All your embedded JIRA issues at a glance.

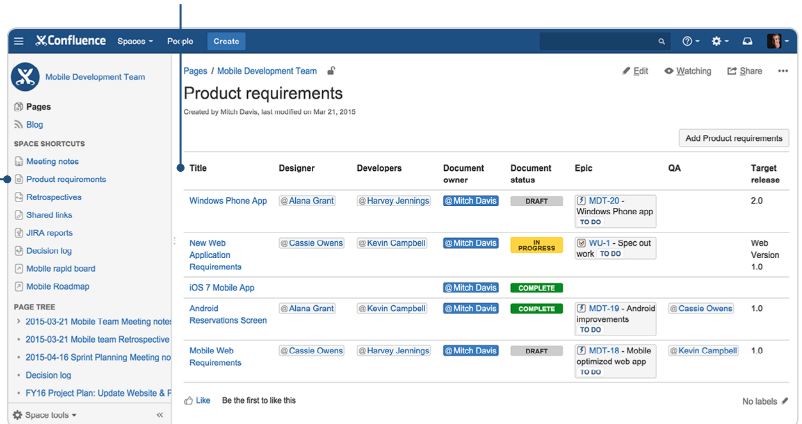



## 8. Your Product Requirements index page

You're probably not just working on one project. Confluence automatically creates an index page that pulls in the key details of all your team's product requirements. You can always access this index page in the *Space sidebar*.

See the requirements your team is working on and their status at a glance.

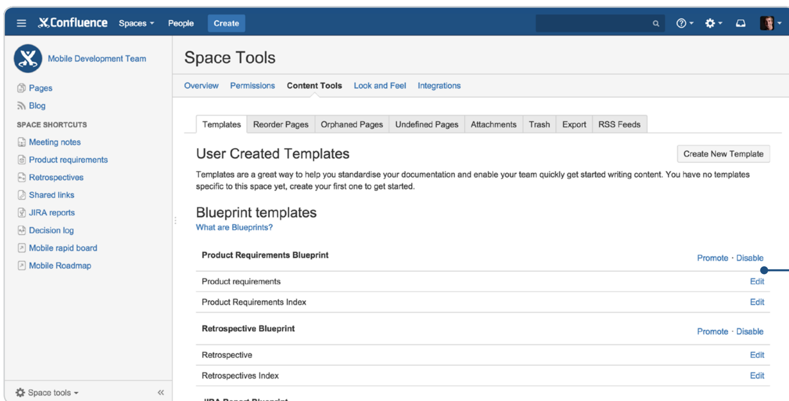
Confluence creates a Space Shortcut for quick access to all your requirements.




It's a handy way to see what your team is or has been working on at a high level. Just like any page in Confluence you can edit this index page as you see fit.

## Bonus: Customize the Product Requirements Blueprint to fit your team's workflow

You can edit the Product Requirements blueprint at any time and tailor it to fit your team's specific needs. If you want to edit the template for your specific team or project, go to the *Space Tools* menu and select *Content Tools*. Under the *Templates* option you can view and edit any of the templates included. If you're a site admin you can also edit the *global template* to customzie it company-wide.




Edit any of the Blueprint templates from the Space Tools menu.

## At a glance: what did you just learn?

The key to agile requirements is buidling a shared understanding amongst your team. The best way to do this is creating your high level requirements doc in Confluence, and collaborating across development and design on crucial issues. Use our template as a guide, and customize it to fit your team or company's workflow.

Here's the completed requirement and a rundown of what we covered:

- **Product Requirements Blueprint:** It does most of the work for you. Follow the page outline, and fill in content that makes sense for your specific requirement.
- **Page properties macro:** This macro makes it possible to create a reporting page across all your requirements. When using the blueprint it will be included on your page.
- **Product Requirements index:** Once you've created a few requirements pages, you can see all the high level details at a glance on an index page that Confluence automatically creates when you use the blueprint.
- **Create JIRA issues:** Highlight text on the requirements page to quickly convert it to a JIRA issue.

- **File comments:** Add your design files to the page and encourage discussion on them with pinned comments.
- **JIRA links:** Just look in the page header to see all the JIRA issues that are included in the requirement.
- **Customizing Blueprints:** You can edit any of the blueprints that come included in Confluence to fit your specific workflow.

Chapter 3

# How to make better decisions as a development team

**By Ryan Anderson**

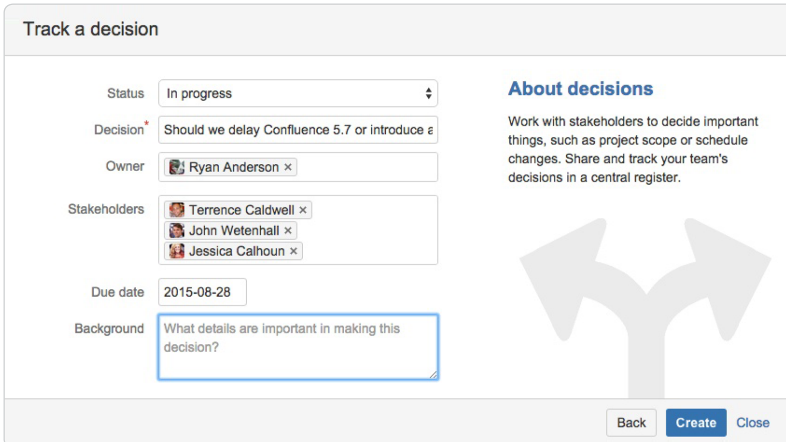
We make countless tough decisions every day at work that take collaboration and careful deliberation—that's what makes them so difficult. Most teams rely on a meeting to make a big decision, but it can be hard to actually make the decision in person. Many times a follow-up meeting is required. Most people are much more comfortable making a decision asynchronously. There's no place better than Confluence.

We have a saying around Atlassian: Decisions don't require handshakes. The Decisions Blueprint provides a simple, repeatable process for making decisions asynchronously amongst your team and recording them forever in Confluence so you, or anyone on your team, can reference them later.

Let's take a look at 5 steps to setting up a decision page in Confluence.

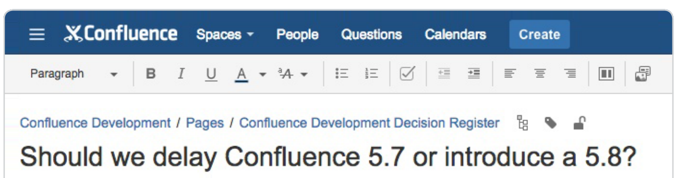
### Step 1. Define: who, what, when.

There are three key components to every decision. What do we need to decide? Who are the stakeholders that need to make this decision? When does it need to be decided by?




## Declare what the group needs to decide

Usually this is best answered with another question. What should we name our new Q&A add-on for Confluence? (Answer: Confluence Questions).

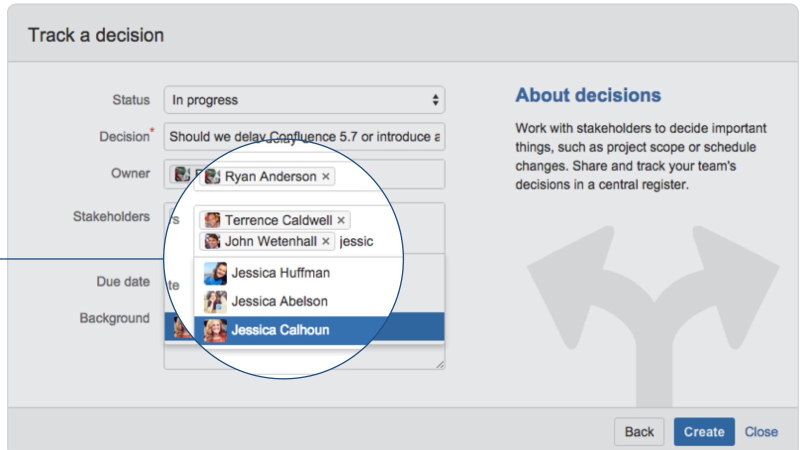



It's best to ask this question in the page title. This makes it a lot easier to find the decision later if you need to search for it.

## Define the stakeholders and the decision owner

With any decision, an often overlooked step is defining roles. There are two groups: The owner and the stakeholders. The owner is the person who has final say, this is likely the page author, but could also be the team leader or executive. The stakeholders are the people that need to be involved in making this decision. This is the pool of people that this decision effects.

Search for other Confluence users to invite them as stakeholders of your decision.

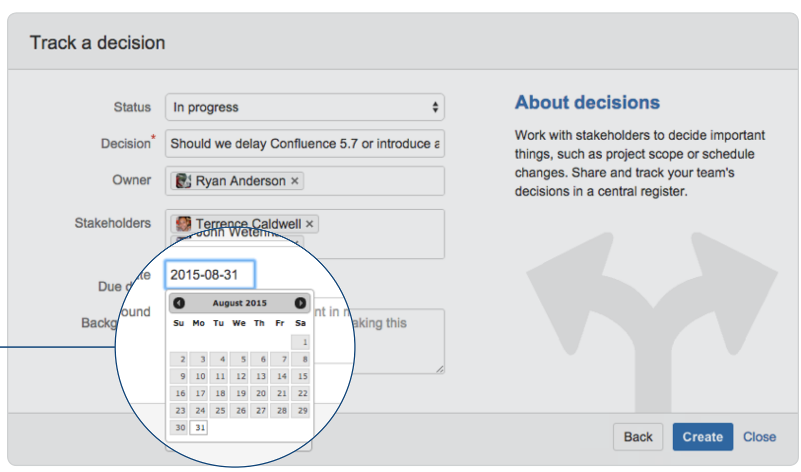



Use @mentions to share this page with the owner and stakeholders. This will automatically send them an invite to contribute—bringing them into the decision page—once the page is saved. This will save you the time of needing to the share the page after you've completed it.

## Define when the decision needs to be made

The thing about decisions is that they have to be made at some point, you can't debate them forever. Set a drop-dead date using the date picker so everyone knows how time sensitive their contributions are.

Use the Confluence date-picker to set the deadline for your decision.




**Pro tip: Skip the 'Background' text field**
I recommend that you use the Confluence editor to write down the background of your decision. This will give you the space and tools to offer a complete context filled with links, images, and even video.

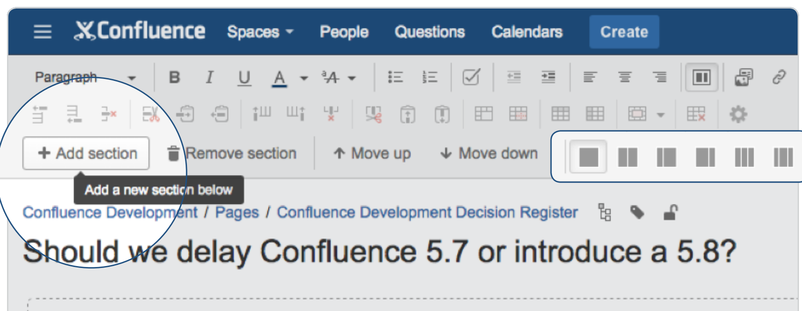
## Step 2. Write: give context so everyone is informed

### Organize

Since you're not going to host a meeting—you need to make sure everyone has all the information before contributing their thoughts. In most cases you might have two options A and B. Use Page Layouts to organize your page and separate your options.

A good tip to working with Page Layouts is click the 'Add Section' button a few times to give yourself some blocks to work with. Place your cursor in the second page layout block and click the two column layout.

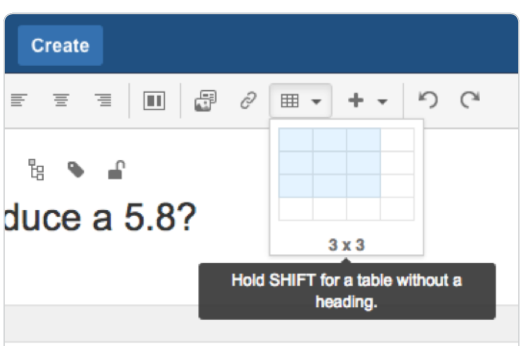
Click this button a few times when you start your page to begin organizing your content.




Choose from the various page layouts to organize the sections on your page

From here you can begin to set up your two decisions. Add a heading to your first column. A quick way to add a Heading 2 is by typing 'h2.'

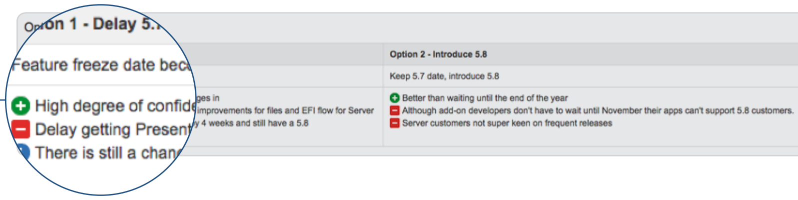
From here, I usually like to organize the options using tables. Tables are powerful tools for organizing information and making your page highly readable.You might want to create a simple pro-con list for Option A. This is simple. Choose a two columns and add 5 or 6 rows. You can always add more later.



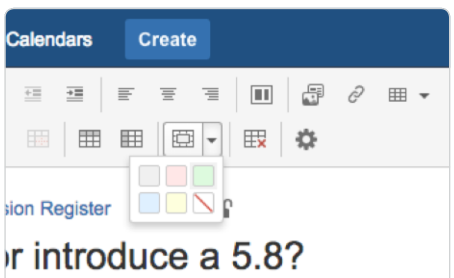
Give each of your table column heading. And then using the ✅ and ⛔ emoticons, you can start creating a list. For a check mark, type '(/)' and for a negative mark, type '(-)'.

Type (+) to add a plus-sign emoticon.

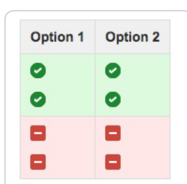
Type (-) to add a minus-sign emoticon.




**Pro tip: Add color to your tables**

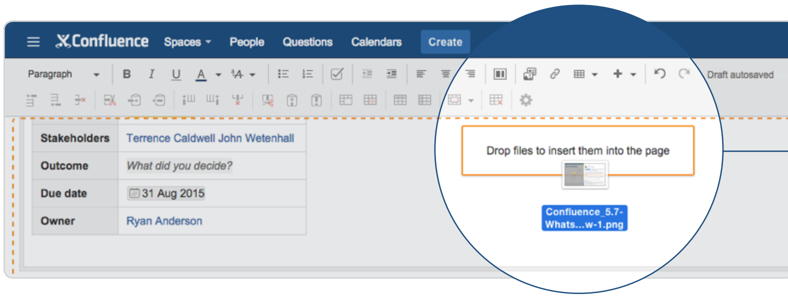



If you want to take your table to the next level, highlight your pro and con list columns.




## Make it visual

There may be decisions you need to make where you want to include some images. This is very simple in Confluence. If the image is on your computer, you can use drag-and-drop—bringing the image inline in the Confluence editor and attaching it to the page.

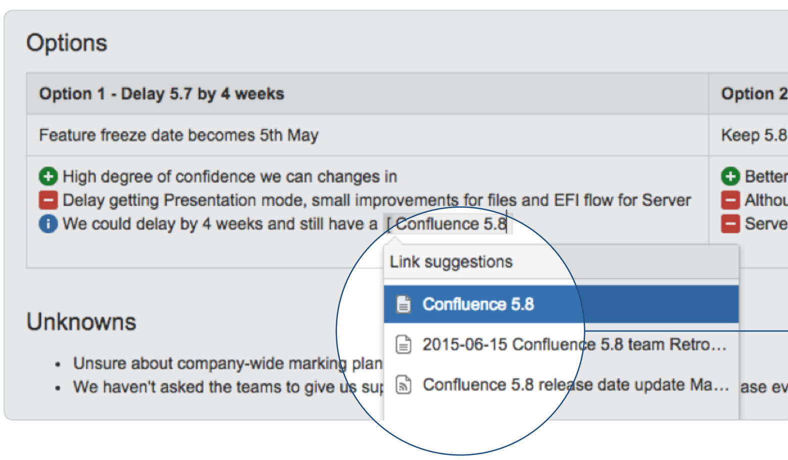



Drag and drop any image or file over the Confluence editor to drop it inline on your page.

## Link to other information

One of the best parts of Confluence is linking. When all of your work and documentation is Confluence, it's easy to link to it and make your teammates aware. This can be really helpful for providing background for a decision. There are a couple ways to link to a page, but in this post, I'll only show you the fastest way which is using autocomplete for links.

Type ' [ ' and you'll see a list of your most recently viewed pages.

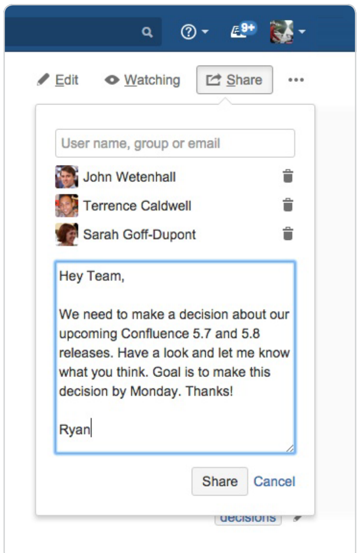



Type [ to trigger autocomplete for links and then search for the page you want to link to.

As you type, you can search across Confluence for the page you want to link to. It's that simple. If you have the Confluence page link already, you can just paste it directly into the editor and Confluence will automatically convert it to a link of your page title.

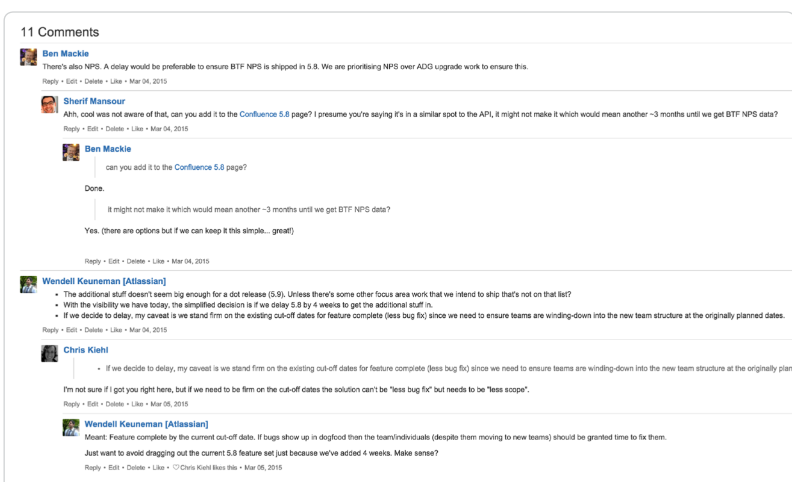
## Step 3. Discuss: hash it out, reach consensus.

You've already shared your page with the stakeholders using @mentions, but if you want, you can share your page again using the Share button. This will send your teammates a notification to visit this page.

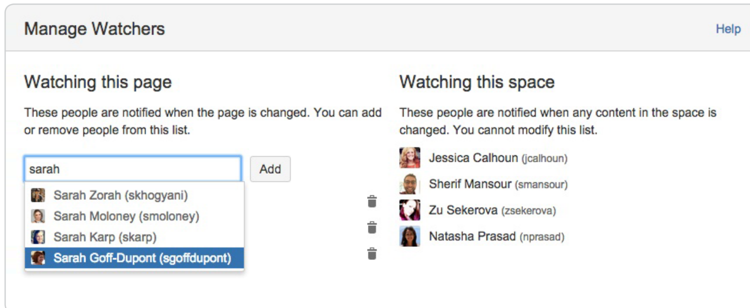



Once you have everyone on your decision page it's easy to foster meaningful discussion. Anyone can comments or like to give feedback, work out details, debate possible solutions, and reach consensus. With all of your conversations in one place, you'll never have to waste time digging through your inbox to find that email thread again.

You can leave a comment at the end of the page:

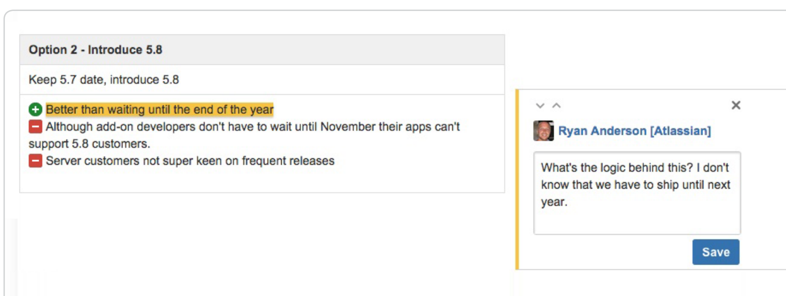


**Pro tip: Add stakeholders as 'watchers' of the page**



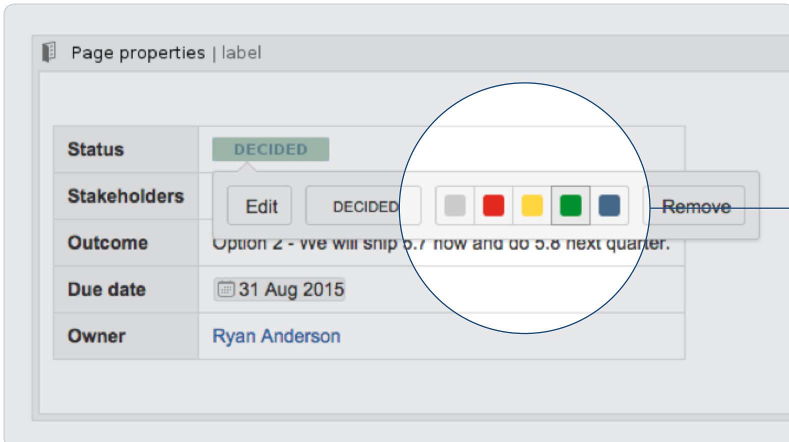

Add your stakeholders and owner as watchers of the page. If they miss the @mention notification (or the Share notification), they will see updates as stakeholders begin to discuss the decision.

Or you can leave a comment inline—typically to seek clarification or point something out in the background that you'd like clarification on. To do this, just highlight any text on the page:




## Step 4. Record: keep everyone informed.

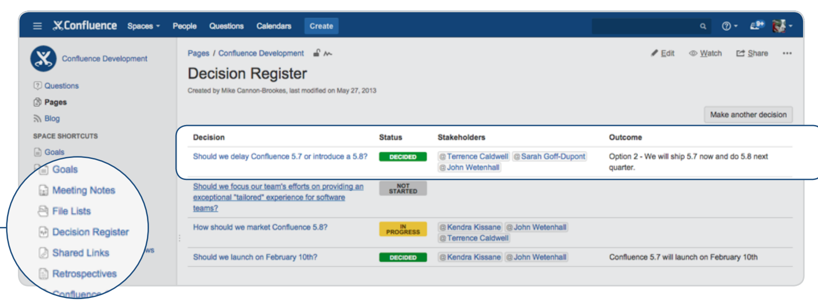
Once your team has reached a decision it's just as easy to record it in Confluence. Change the status of the decision, using the Status Macro, so that everyone that contributed is notified. With everyone on the same page, it's easy to keep progress moving forward.




Change the color of the Status macro to green to show that a decision has been reached.

## Step 5. Collect: every decision in one place.

Want to refer back to see why a decision was made? The Decision Blueprint automatically indexes your decisions on a single page. Now all your decisions are accessible to anyone who needs them, whenever they need them.




Access all of your team's decisions from the space sidebar.

Access all of your team's decisions from the space sidebar.

## At a glance: what did you just learn?

Creating this page might seem daunting with all the macros, but making informed decisions as a group is worth the work. It gives your team one place to discuss your options and save yourself long meetings and even longer email threads. In the long run, you'll be happy that you can make a decision once. Here's a list of what we covered:

- **Page layouts:** Organize your page with different column layouts.
- **Emoticons:** Use the ✅ and ⛔ emoticons give your page some color.
- **Tables:** These are awesome tools for organizing your thoughts.
- **Adding Images:** Drag-and-drop is the fastest way to insert and image or file into a Confluence page.

- **Linking:** With all your work in Confluence, it’s easy to link off to relevant pages using Autocomplete.
- **Comments:** Leave your overall comments at the end of the page and your targeted comments inline.
- **Status macro:** Stamp your decision made by updating the color and text.

Chapter 4

# How to create your team's technical and onboarding documentation

**By Terrence Caldwell**

Successful dev teams need to be able to move fast, without sacrificing code quality. Since you can't just go out and make clones for your developers (yet...), how do you make your team faster?

Try these two tactics: first, speed up the time it takes to onboard new team members. Second, limit the amount of time your team spends searching for answers to technical problems.

Of course, you want to spend your time building awesome products and not writing docs. At the same time, you need good docs to help your team build awesome stuff. Confluence helps solve the paradox by making it easy to create and document your standard technical docs so everyone can access and contribute to them. In short order you'll be a lean, mean documentation machine.

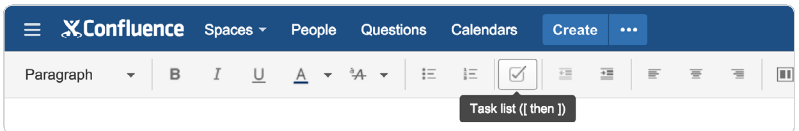
Check out three standard technical docs that your team can create in Confluence:

## 1. Onboarding documentation

Getting new people current is critical to your dev speed as well as the future success of your team.

### The new developer checklist

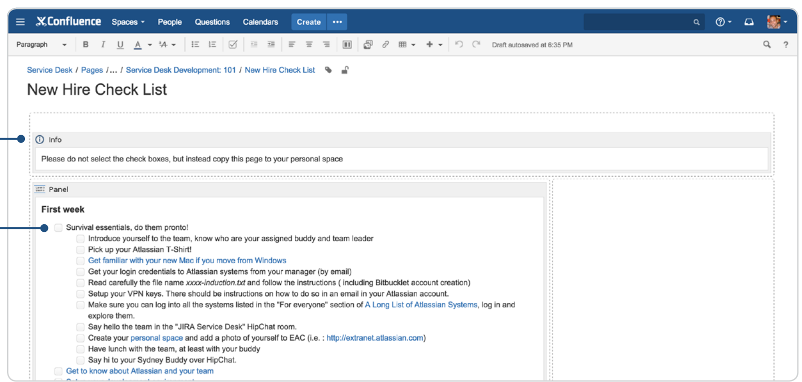
Start new hires with a guided task list. You can add tasks to any Confluence page from the editor toolbar:




The new hire task list should provide direction and useful resources. Include instructions on logging into critical systems and applications, links to important internal resources like company policies, and tasks for meeting members of the team.

Use the info macro to add a callout or instruction.

Create tasks for recommended reading, important links, and to-dos.

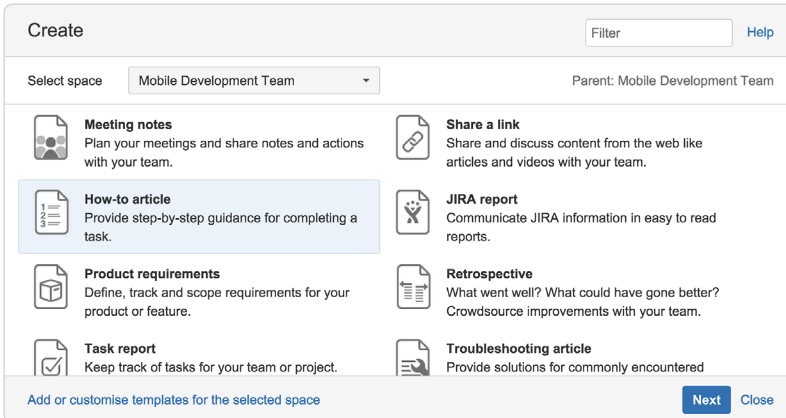



**Pro tip:** Create one universal new-hire task list that lives in your team space. Whenever a new person starts have them copy that page and move it into their personal space. They can edit the task list to add additional tasks, as well as assign themselves due dates for tasks to be completed.

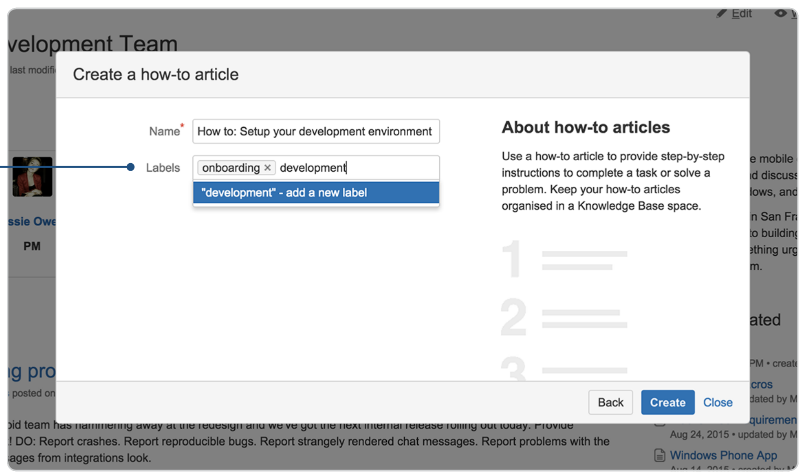
## How-to documentation

Your task list is a great resource for pointing people in the right direction. But what are you pointing them to? You’ll need a handy set of how-to docs to help new hires help themselves.

Get started with our how-to blueprint—it gives you the basic layout so you can focus on adding content.



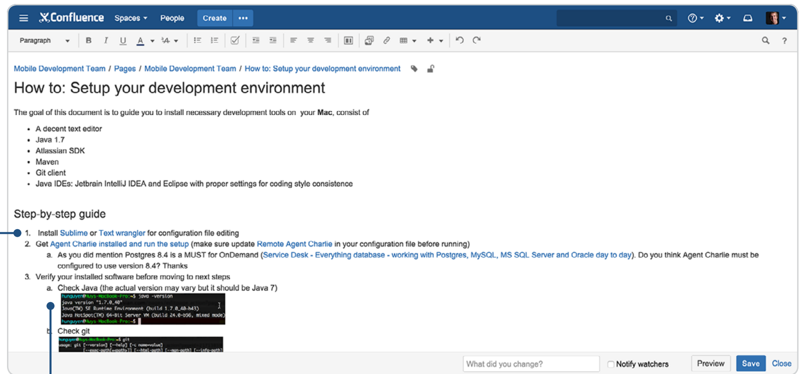

Keep your onboarding docs organized by adding descriptive labels.




**Pro tip:** Label all your onboarding docs with the same label, like “onboarding”, or something similar. With all your docs labelled, you can create a central repository of them using the content by label macro.

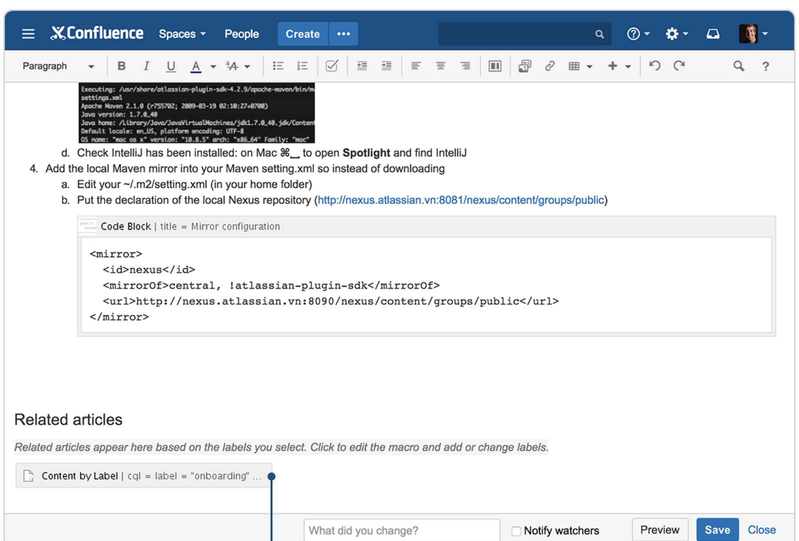
Make sure to add images, code blocks, and links to relevant web content to your page.

The how-to blueprint includes a bulleted list for step-by-step instruction.




Drag and drop images into the page to add visual guidance. These are screenshots taken from an IDE.

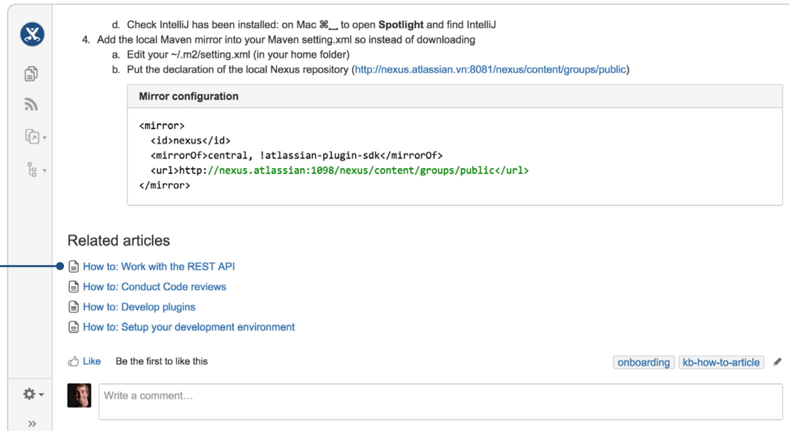
After you've completed your page, you'll notice the *Content by Label macro* at the bottom. This macro automatically pulls in all the pages in the current space with the same label.




The content by label macro makes it possible to display other pages with the same label.

Once the page is saved, you can see a dynamic list of pages with the same label:

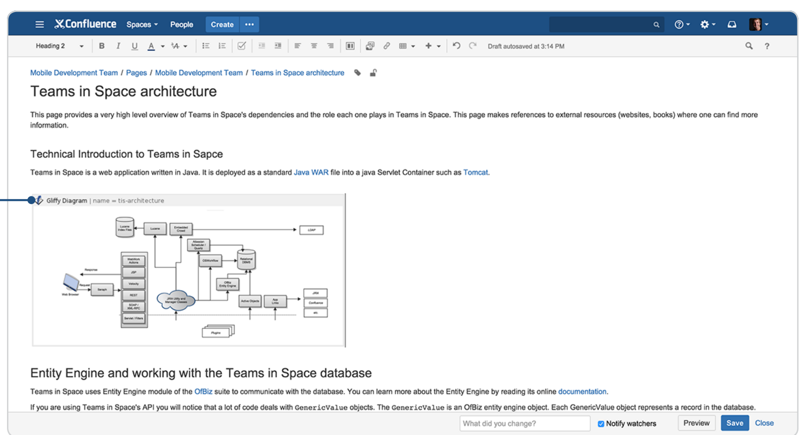
These pages all have the "onboarding" label and are automatically added to the bottom of the page.




## 2. Application and network architecture documentation

These documents need to be accessible to the whole team, at all times. Here's how to put together your architecture docs in Confluence:

Use diagramming tools like Gliffy to embed rich diagrams that are easy to edit.

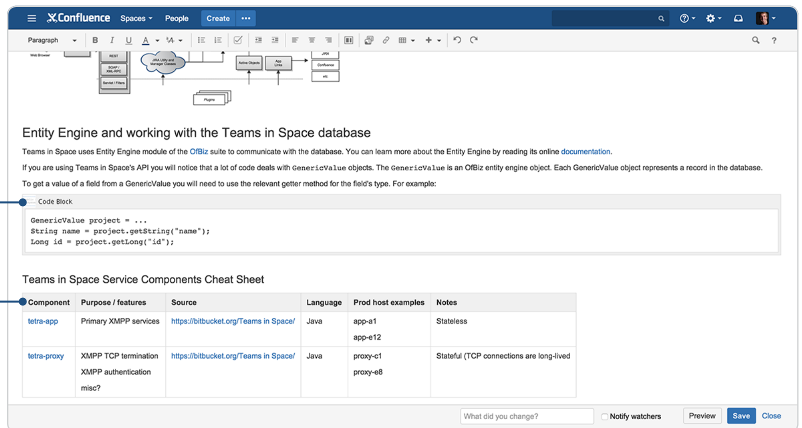



**Pro tip:** Use diagramming tools to embed architecture diagrams. There are two great add-ons available in the Atlassian Marketplace that are perfect for this: Gliffy and Draw.io. Both these add-ons give you a diagram editor right within Confluence. We use these tools as well as websequencediagrams.com for simple relational diagrams that you can attach as images.

The most important part of your architectural docs is being able to reference diagrams, components, SDKs, APIs and related docs all in one place. These documents can be as heavy or light on information as you need. At Atlassian, we put all this information in tables. Tables are easy to insert into your page, and flexible. Use the table tools menu to add and cut columns and rows, merge, and highlight cells.

Use the code block macro to refer to code snippets.

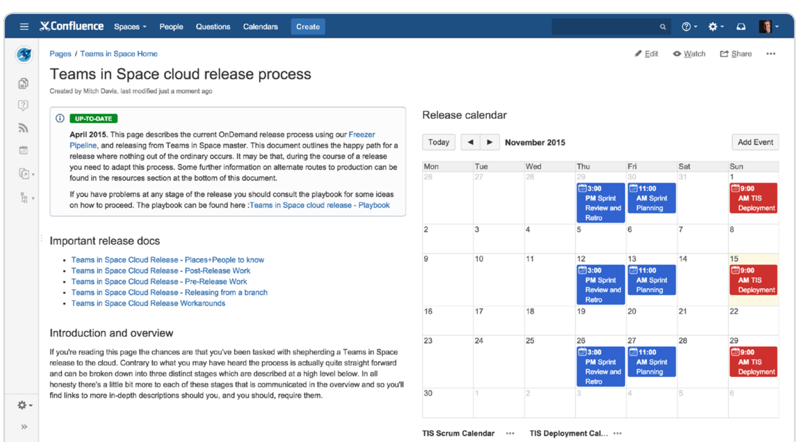
Use a table to list components along with important links and notes.




Just like your onboarding content, you'll want to label all your related architectural docs to keep them organized.

## 3. Release process documentation

The other important set of documentation you'll want to create and store in Confluence is your release process and readiness docs. These are the documents that streamline your releases and help build a culture of continuous improvement, and are probably constantly evolving as you adopt new practices to improve your process. So you'll want to make sure everyone has access to the latest 'n' greatest—which is exactly what Confluence was built for.



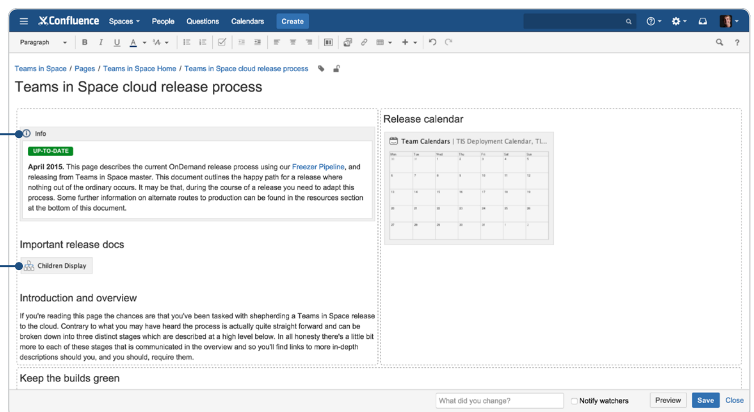
There are three key elements to your release process documents:

1. **Continuously refresh and improve:** Every release should serve as a learning process for your team. As you improve your development velocity and release cadence, you'll want to keep your docs up-to-date. We use the info macro as a visual indicator of the page's relevance.

2. **Centralize related docs:** You'll want to document the relevant stakeholders, potential workarounds, and all the necessary comms that go out before and after a release. At Atlassian, we use separate child pages for all of this information. To show all of these related pages, add the children display macro to the top of your page. This macro shows all the subpages of your homepage, so you have quick access to the page you need.

Use the info macro for important callouts and to show if the doc is still up to date.

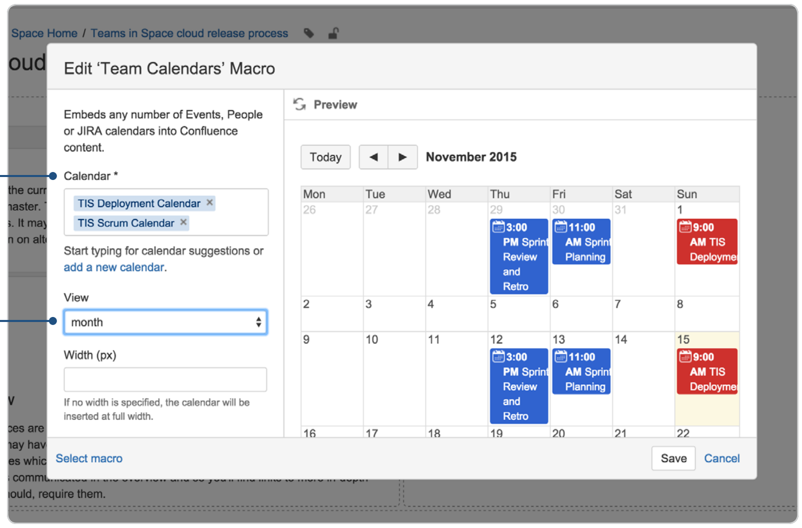
Show all the related child pages of your main release process page.




3. **Visualize the process:** Team Calendars is an add-on for Confluence, made specifically to help teams track team leave, JIRA releases, and events in one place. Embed a calendar on your release readiness page to give everyone visibility into your release dates and avoid conflicts by seeing the whole team's availability on those dates. You can view by month, week, or on a timeline.

Choose the calendar(s) you want to embed with the Team Calendars macro

Select how you want your calendar to be displayed on the page




## At a glance: what did you just learn?

Having a rich, reliable, centralized set of technical documentation might not be the sexiest aspect of your development team, but it can help your team move faster and go from good to great. Here's a quick recap of what we covered:

- **Page tasks:** Create an onboarding task list for all new hires.
- **How-to article blueprint:** Use the included how-to article page blueprint to quickly document important development how-to guides.
- **Content by label macro:** Group all the pages with the same label.
- **Diagramming add-ons:** Create, edit, and share diagrams right within Confluence.
- **Code block macro:** Embed code snippets from any programming language into your page.
- **Children display macro:** Show all the child pages that are organized under your parent release process page.
- **Info macro:** Add a visual callout at the top of your page.
- **Team Calendars add-on:** Visualize release dates, team leave, and other events on one calendar.

Chapter 5

# How to build a release planning page

**By John Wetenhall**

Every product release requires a lot of hard work and a ton of coordination between individuals and teams. In some places, a release coordinator takes this on as a full time job, while in others, a developer, product manager, or project manager might be in charge of release planning.

At Atlassian, we use Confluence to collect and organize all the work and information for each release, and to communicate what it's all about to the rest of the business.

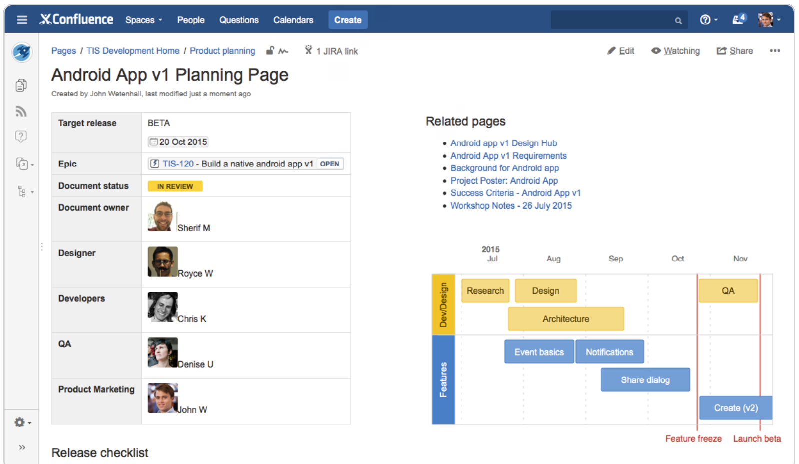
I've written before about how we approach releases, what goes into them, and how we organize them. In this post I'll cover exactly how to build an awesome release planning page that will help you do the following:

- **Identify** the key objectives, people, and high level roadmap
- **Collect** all the relevant background information and ongoing work
- **Direct** your colleagues to additional details in JIRA and on related pages

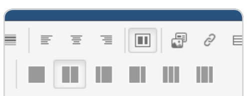
You can set your release up for success from the start with a single source of truth for everyone working on it. Follow these steps to create an awesome release planning page.

## Step 1: Identify the key details

It's essential that anyone who comes to your release planning page can quickly identify who is involved, and what the goals and expected outcomes are. When you're done, the top portion will look something like this:




Start by creating a blank page and then add a two column page layout so you can fit all your key information above the fold. *Page layouts* are in the editor toolbar here:

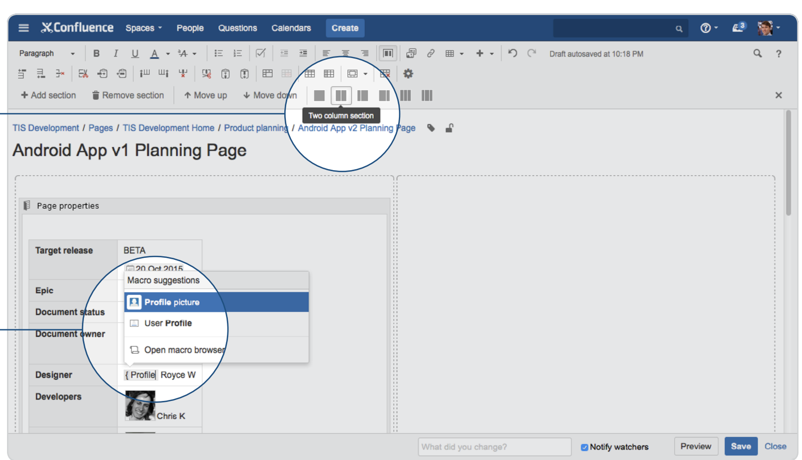


## Who?

Add a table inside it that displays the high level details of your release. We like to use the profile picture macro, which you can find in the macro browser or by pressing { and then typing "profile picture", to display each person's image. This helps people put a face to a name, especially when we're working in remote teams. Hover over someone's name in an @mention and you'll find a bit more information about that person.

Give your page some structure with a two-column page layout.

Use the Profile picture macro to help colleagues put a face to the name.



> **Pro tip:** Put your table inside a *page properties macro* so you can create a report with a summary of the key details of all your releases available for easy reference.

### What?

You'll also want to capture critical details like the name of the release, expected ship date, and the status of the planning page or even the release itself. We use the status macro to communicate status of pages or individual line items. You'll find it in the "**Insert more content**" button in the editor toolbar or by pressing {. Change the text and the color to indicate changes in status.

Example planning page statuses: DRAFT IN REVIEW CURRENT

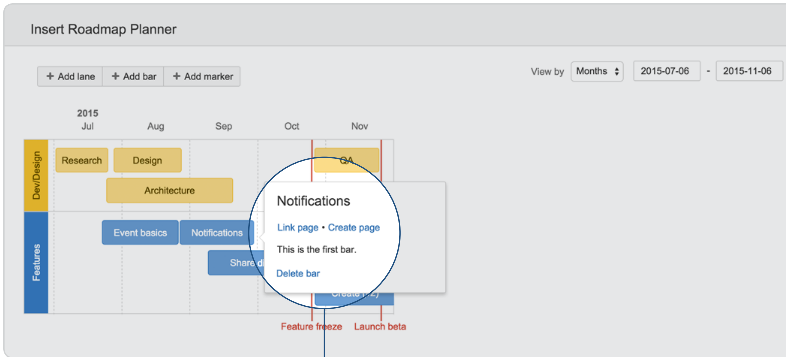
Example release statuses: PLANNING IN PROGRESS AT RISK SHIPPED

We also include the most important goals for the release as well as the non-goals—i.e., things that were discussed that we've actively decided not to include in the scope of the release. It can be helpful to link to decision pages or others with background on how the team established the goals.

## Step 2: Visualize your plans

We like to sketch a roadmap to get a rough idea of how various streams of work will fit together and to communicate the timing within the team and to others. We use the roadmap macro in Confluence to visualize our plans at a high level. (Note: when we want to create a data driven roadmap and determine what's possible given our resources, we use Portfolio for JIRA).

Insert the *roadmap macro* just like any other macro. You can then create as detailed or as simple a roadmap as you need by adding additional lanes for work streams, bars for epics, and markers for milestones (or whatever combination works for you!). Move the bars around to experiment with different timing, and drag them to resize.




Link to other Confluence pages or
add a description for more detail.

Want to give people a little more context on a particular epic or feature you've listed as a bar? You can link any bar to a page—we often provide a link to a product spec that gives more detail. Click insert to add the roadmap to the page, and remember that if your plans change, your roadmap can, too, just edit it and move things around as necessary.

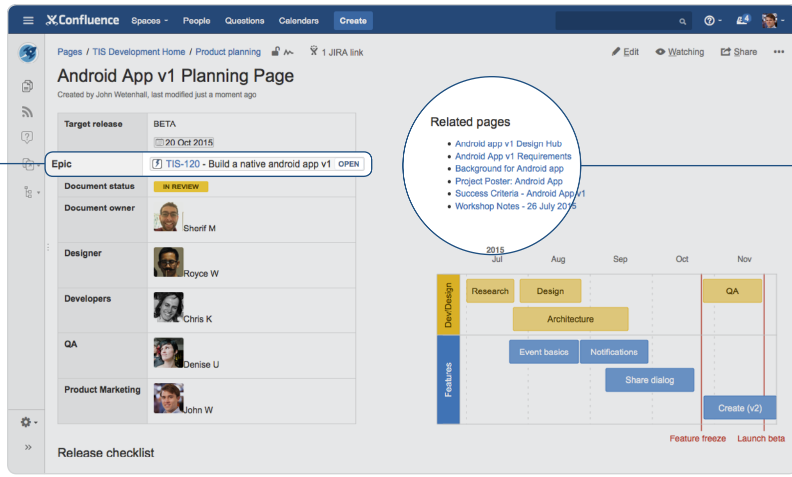
## Step 3: Connect the dots

The final piece in your release planning page is all about connecting people to the information they need. You can do this in whatever way best suits your release process, but there are a few basic tips.

First, link your page to the JIRA epic where you're tracking the work for this release by simply pasting the JIRA link into your table at the top. If this release requires multiple epics, you might want to include them in a table below the fold. Paste the issues in so you get the latest status, dynamically updated.

Second, add the children display macro at the top right of your page, above your roadmap under a heading that says "Related Pages". This displays the child pages (sub pages, if you will) of the release planning page. You can specify how many pages and what content to show in the macro editor.

Paste a link to the relevant JIRA epid to connect your plans with where you track the work.




Group related pages as children of your planning page and display them with the children display macro.

Encourage your colleagues to create any page that has to do with the release as a child page of the planning page so they're all collected in one spot. You may even want to move other pages to become child pages if they were created before the planning page existed. You can move a page from the "…" dropdown in the top right corner.

Pages that we typically create for a release include the following:

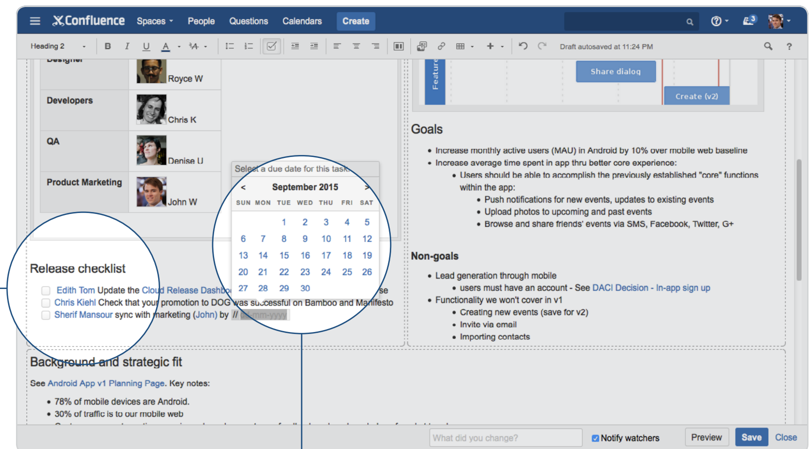
- **Design Hub:** for all the relevant UX/UI designs
- **Competitor insights:** an overview of how other tools tackle this problem
- **Analytics:** how will we measure usage? what other feature usage might be affected?
- **Success criteria:** what metrics do we want to hit?
- **Workshop notes:** takeaways and whiteboards from related spikes and workshops
- **User testing:** plans and notes around testing the new features with users

Finally, we like to add links and context below the fold so anyone can get to the additional information they need. Things we typically include:

- Summary of the problem we're solving for the customer
- Links to related feature requests in our public issue tracker, JIRA
- A high level view of the epics required for this release
- A list of open questions regarding the release

We finish off the page with a release checklist using tasks. Create a task list with the check box in the editor toolbar, or type a left bracket "[" then a right bracket "]" to create a task. You can assign tasks with an @mention and add a due date by typing "//". This lightweight task list makes sure we don't forget any of the little things before we release.

Assign tasks right on the page for a lightweight release checklist.



Type "//" to set a due date for a task.

The end result is a page that covers what you plan to release, why, who is involved, an estimate of the timeline and links to more information for anyone who needs it.

## At a glance: what did you just learn?

Creating this page might seem daunting with all the macros and information we suggest including, but once you create one, you'll get the hang of it. Here's a list of what we covered, plus links for more detail:

- **Page properties macro:** put the table at the top inside one of these to make collecting release pages easy.
- **Page layouts:** Organize your page with different column layouts.
- **Profile picture macro:** Helps put a face to a name, especially in distributed teams.
- **Status macro:** Communicate status or use it as a colorful label.
- **Roadmap macro:** Sketch a visual roadmap with just a few clicks.
- **JIRA links:** Connect your page in Confluence with your project and epics in JIRA.
- **Children display macro:** Expose related pages and encourage organizing all release information under the planning page.
- **Task lists:** Assign tasks with due dates to make sure the little things get done.

Chapter 6

# How to create sprint retrospective and demo pages (like a BOSS)

**By Sarah Goff-Dupont**

*"Knowledge un-shared is knowledge wasted."*

*– this really smart guy I work with*

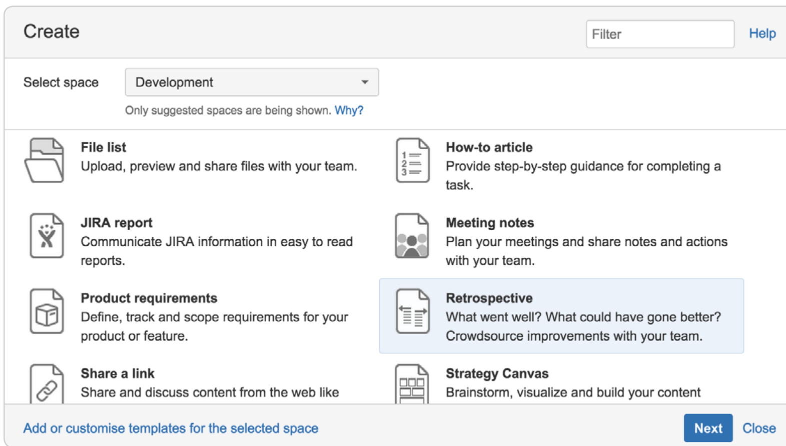
I mean it in the most respectful way possible when I say that we software makers are inherently lazy. Else, why would we choose careers making computers do repetitive tasks for us? It's this laziness that also motivates us to continually improve the way we work and the products we make—we want life to be easier for both ourselves and our customers.

But learning and improving is really hard if you have to learn the same things over and over again. Even the ancient humans knew this, so they got into the habit of writing stuff down (proving that software makers don't hold a monopoly on strategic laziness). Fortunately for *homo modernus*, we can skip the carving of stone tablets and use Confluence instead.

As software makers, we rely heavily on sprint retrospectives and demos to learn more about what's going on with our team and our products. Read on to see how to build pages that record what you've learned and let you share it outside your team.

## Sprint retrospectives

Confluence gives you a big leg up with the Retrospective Blueprint. If you've never used a blueprint before hit the ellipsis next to the *Create* button at the top of the screen, select *Retrospective*, then hit the *Next* button. You'll be prompted to choose a title for the page and indicate which team members participated—they'll be @mentioned on the page, which means Confluence will share the page with them automatically.

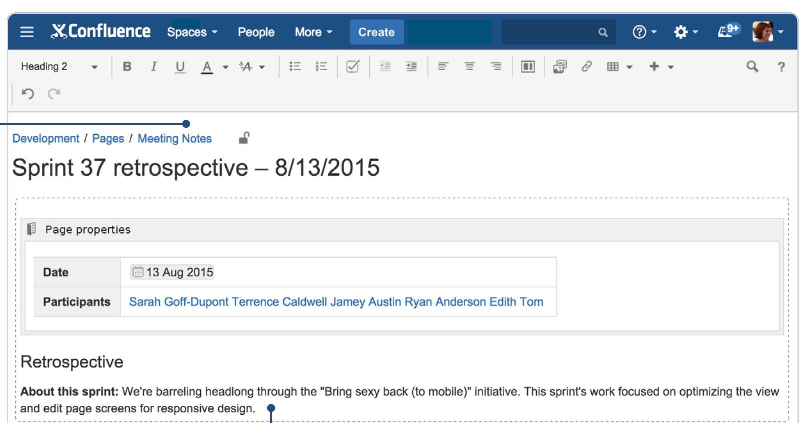



Once that's all set, click the *Create* button at the bottom of the dialog to fire up your new page. You'll notice that the blueprint divides the page into sections with different layouts—full-width for the top and bottom sections, and a two-column layout for the middle one.

## Set the stage

Although the blueprint doesn't call for it, I like to add a sentence or two describing what our focus was for this sprint to help jog our memories when we come back and look at the page a month (or year) later.

Organize all your meeting notes under a parent page so they're easy to find.

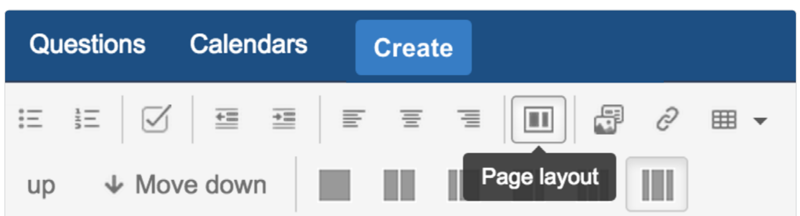



A quick note about the sprint provides just enough context around what's on the rest of the page.

## Capture your ideas

The Retrospective Blueprint defaults to a “what went well / what didn’t go well” format, which allows the whole team to help translate one person’s feedback into actionable tasks. If your team is more experienced, you can easily adapt the blueprint to use a “start / stop / continue” format—either instead of or in addition to the default sections provided.

To split that section of the page into three columns, place your cursor in one of the two existing columns, click the *Page layout* button in the editor, and select the three-column option. Then copy the contents of an existing column into your new third column and update all three headers appropriately. And if you use both mechanisms in your retrospective, add a new section to the page by clicking the *Page layout* button, then Add section, and choose the three-column layout for the new section.

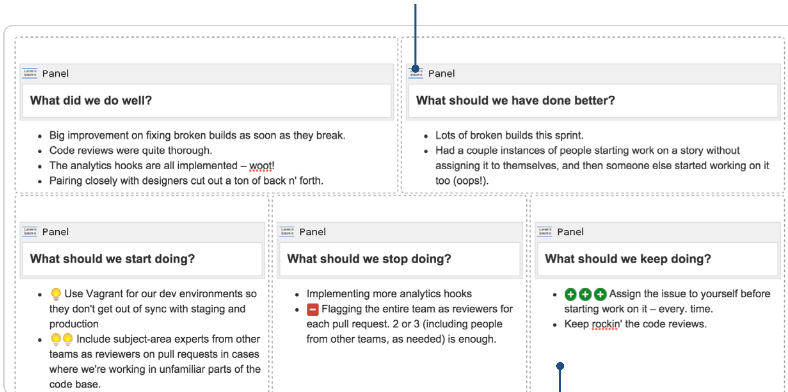



Now you’ve got an easy way to capture and categorize the things you and your team discuss. But how do you know which are the most important points—the ones you want to make a priority in your next sprint?

At Atlassian, our teams put it to a vote. Each team member gets to up-vote 3 items in each category, which we represent with an emoticon placed next to the item. Here are some suggestions:

- : keystroke is (+)
- : keystroke is (-)
- : keystroke is :-)
- : keystroke is :-(
- : keystroke is :-P
- : keystroke is (y)
- : keystroke is (n)
- : keystroke is (on)

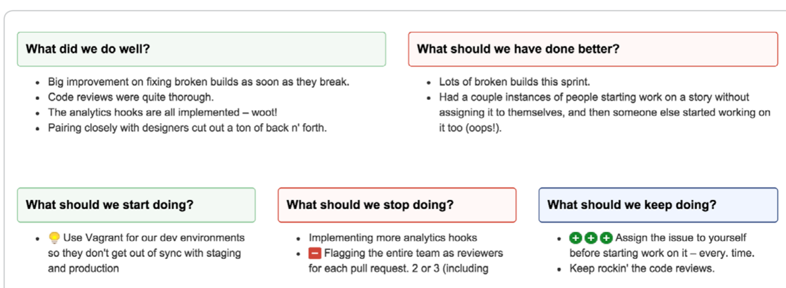
Panels have red and green backgrounds by default—you'll see the colors when you preview or save the page.




Mix and match different layouts for each section of the page.

**Pro tip:** You can change the background color of a *Panel* macro by clicking the panel and selecting *Edit*. I like a nice, calming, everything's-cool blue background when adding a "keep doing" section. Try #334C80 for *Border Colour* and #F0F5FF for *Background Colour*. All you need is the hexidecimal color codes to customize the look of your panels.

Once you save the page, those panels will look something like this:

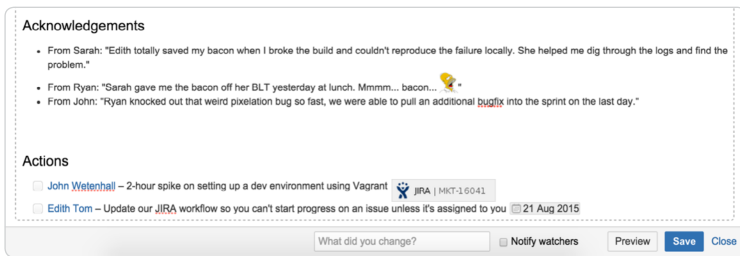



## End on a high note

Another habit of highly-successful teams is for team members to acknowledge each other for the good deeds they did this sprint. (Note: managers are barred from giving or receiving acknowledgements—this is meant to be strictly peer-to-peer.) Above the "Actions" section created by the blueprint, type "Acknowledgements" and format it as a section heading by clicking on the *Paragraph* menu, and selecting *Heading 2*. Underneath, transcribe the kudos your team gives each other in a bulleted list.

Finally, don't let your great ideas go to waste—take action. The blueprint adds a section for this automatically and gets you started with an empty checklist. Three formatting features come in really handy here:

- @mentions: so you can flag a team member as owning each item
- Dates: so you can set deadlines for following up (type // to bring up the date picker)
- JIRA issues macro: so you can see the progress of larger action items (type { then JIRA to create a new issue without leaving Confluence)

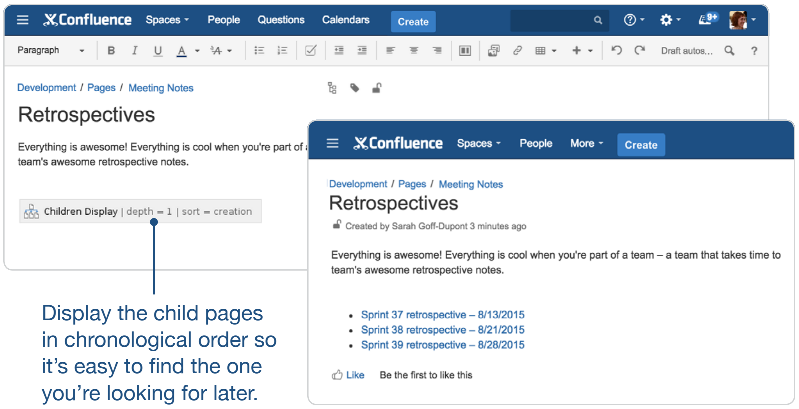



## Sprint demos

You probably looked at this headline and thought "Huh?!" No worries—we tend to think of demos as happening live and in the flesh (or at least in real time). But for distributed teams, that's not always possible. (And let's face it: distributed teams are as ubiquitous in the software world as Elvis is in Las Vegas.) So when you can't get everyone online at the same time, recording demos using screen capture tools like Camstudio or Jing and embedding the videos on a Confluence page is a great substitute for doing it live.

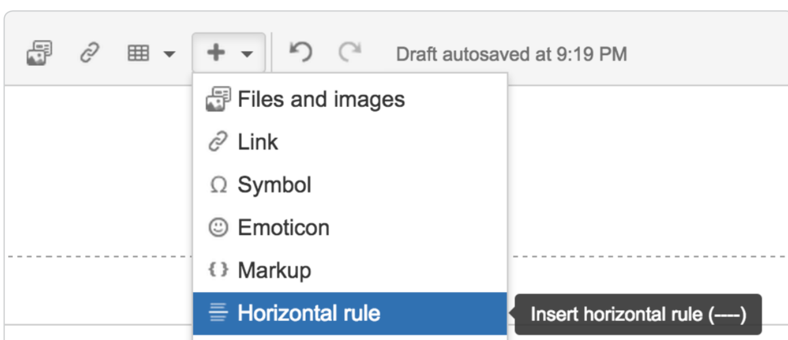
### More fun with page layouts

Start by creating a good old fashioned blank page and give it a sensible title that'll make it easy for team members to find through Confluence's quick search or by browsing your team's space. For example, you might create a parent page called Retrospectives, then child pages underneath for each sprint's retrospective notes.

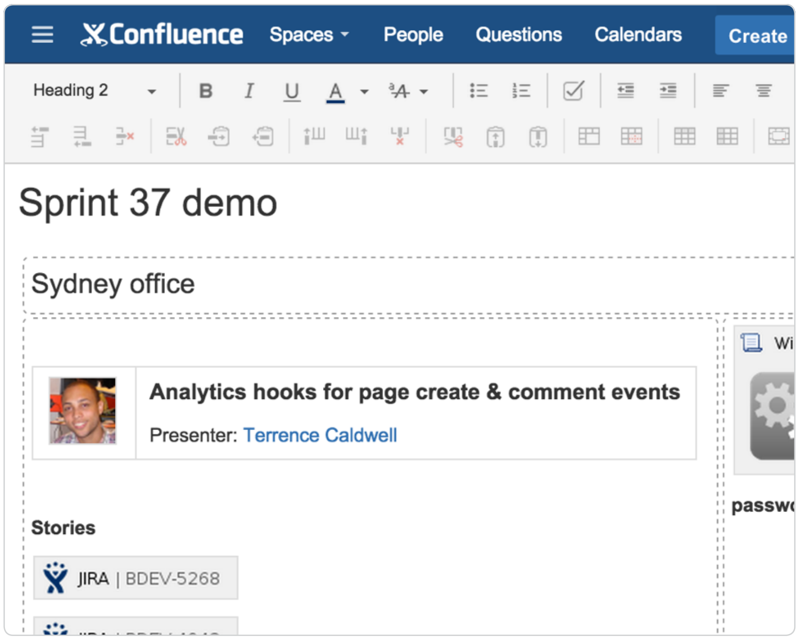



Bonus points if your parent page uses the *Children display* macro (which does not actually crawl Facebook and display pictures of your children on the page—because that would be creepy). Anyway...

Wherever you choose to create the page, you now get to use your new friend, the *Page layout* button, to add sections for each office or locale. I like to use a full-width (single-column) section for the headings, and insert a grey line that extends all the way across. To add it, go to the *Insert more content* menu and select *Horizontal rule*.




For the sections containing the demos themselves, use a two-column layout to keep the page's content compactly organized and easy to read. In the left-hand columns, insert a table to hold information about who is presenting and the theme of their demo. Below that, embed the relevant JIRA issues using the *JIRA* macro, or simply by pasting in the link to each issue, as I mentioned above.

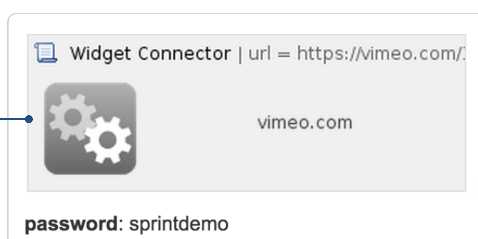


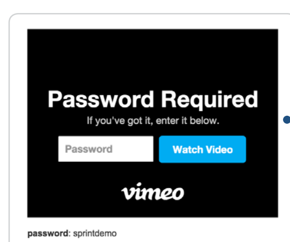

**Pro tip:** This Profile picture macro is perfect for placing in each section. It's just nice to be reminded of what your team mates look like when you don't see them every day.

## Fun with video

Over in the right-hand columns, ask each team member to embed their screen captures using either the Widget connector or Multimedia macro. Both are available from the *Insert more content* menu—choose *Other macros* at the bottom of the menu to bring up the macro search and selector screen.

Widget Connector, as seen when the page is in edit mode.

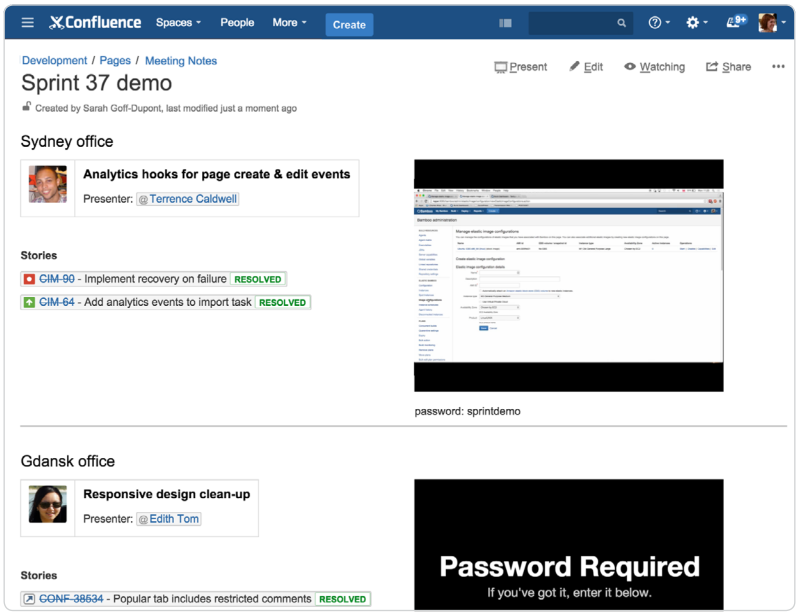



Widget Connector, as rendered when the page is in view mode.

**Pro tip:** If you're freaked out by the idea of putting videos of your un-released work out on the internet where users and competitors could discover it, you're not alone. Try creating an account for your team on Vimeo—it's a lot like YouTube, but they make it super easy to add password protection to your videos.

Once you've saved the page, it'll look something like this:




## At a glance: what did you just learn?

We've covered a lot of ground here: blueprints, and page layouts, and macros (oh my!). Here's the highlight reel.

- **Retrospective Blueprint:** It does most of the work for you. Just fill in the blanks, add optional sections for acknowledgements and sprint overview, and hit Save.
- **Page layouts:** Divide all or parts of your pages into one-, two-, or three-column layouts to make your pages look even more awesome. Careful, though: this feature is highly addictive!
- **JIRA links:** Paste in a link to your JIRA issues, or pull up the JIRA macro to search for them.
- **Profile picture macro:** Because your team mates look great in pictures.
- **Widget connector macro:** Paste in YouTube, Vimeo, Twitter, or SlideShare links, and this macro will make your content look great on the page.
- **Multimedia macro:** Use this to upload .mp4 video images directly from your computer. Or .mp3 files if you care to provide a soundtrack for demo viewing.

# How to use blogs to share your development team's progress

**By Ryan Anderson**

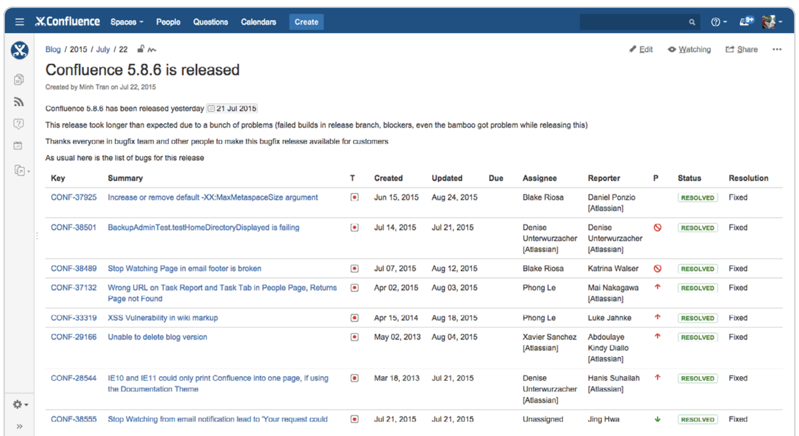
One of the hardest things to do as a development team is keep everyone in your team and the organization up to date. Some people want to know the status of "Project X", others want to know what new features are coming out and when, and still others want to know what's coming next quarter. Many teams rely on big state of the union meetings or mass emails, but neither option is effective when it comes to transparency or visibility. People miss emails, and emails get buried. After a big announcement, there's no way to reference things said. Plus, there's no simple way to ask questions.

How about a Confluence blog post? In this article, I'll show you how to utilize blog posts to share important news and updates across your development team and greater organization. Blogs are a great tool for keeping teammates and coworkers updated because they don't get buried in all of your email, you can write them using dynamic content like JIRA issues and macros, and they're social which makes it easy to start a conversation with interested stakeholders without a never ending thread of emails. Check out these 3 blogs we publish in Confluence all the time at Atlassian.

## Blog 1: Project and release updates

At Atlassian, before a new release goes out to customers, a member of our development team writes an internal blog post in Confluence to update Atlassians of what's in the release. Our developers want to know what features shipped and what changes to the products were made. It’s the easiest and most effective way to keep everyone up to date.

The easiest way to showcase which features shipped is to bring in issues from JIRA using the JIRA Issue macro, which embeds a single JIRA issue or a list of issues in a Confluence page. The embedded issues are dynamic, so anyone can click through to JIRA to get more details about what shipped.




Confluence 5.8.6 is released

Created by Minh Tran on Jul 22, 2015

Confluence 5.8.6 has been released yesterday 21 Jul 2015

This release took longer than expected due to a bunch of problems (failed builds in release branch, blockers, even the bamboo got problem while releasing this)

Thanks everyone in bugfix team and other people to make this bugfix release available for customers

As usual here is the list of bugs for this release

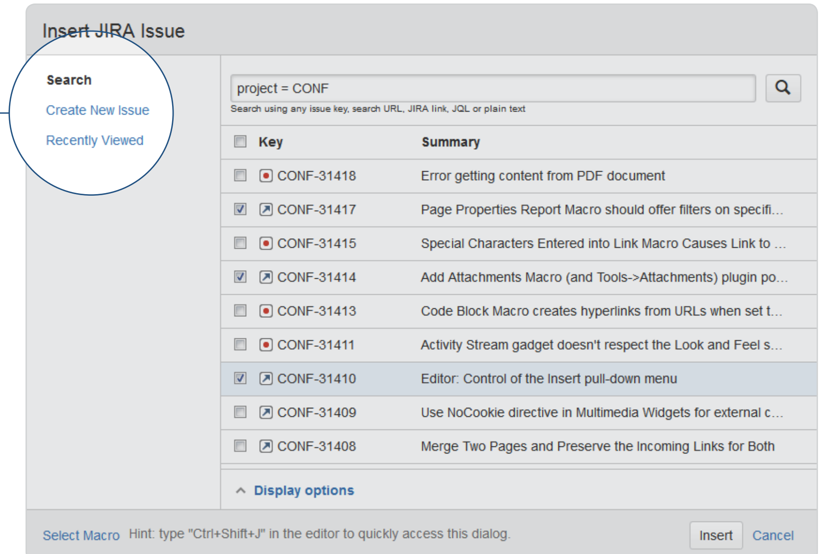
| Key | Summary | T | Created | Updated | Due | Assignee | Reporter | P | Status | Resolution |
|---|---|---|---|---|---|---|---|---|---|---|
| CONF-37925 | Increase or remove default -XX:MaxMetaspaceSize argument | | Jun 15, 2015 | Aug 24, 2015 | | Blake Riosa | Daniel Ponzio [Atlassian] | | RESOLVED | Fixed |
| CONF-38501 | BackupAdminTest.testHomeDirectoryDisplayed is failing | | Jul 14, 2015 | Jul 21, 2015 | | Denise Unterwurzacher [Atlassian] | Denise Unterwurzacher [Atlassian] | | RESOLVED | Fixed |
| CONF-38489 | Stop Watching Page in email footer is broken | | Jul 07, 2015 | Aug 12, 2015 | | Blake Riosa | Katrina Walser | | RESOLVED | Fixed |
| CONF-37132 | Wrong URL on Task Report and Task Tab in People Page, Returns Page not Found | | Apr 02, 2015 | Aug 03, 2015 | | Phong Le | Mai Nakagawa [Atlassian] | | RESOLVED | Fixed |
| CONF-33319 | XSS Vulnerability in wiki markup | | Apr 15, 2014 | Aug 18, 2015 | | Phong Le | Luke Jahnke | | RESOLVED | Fixed |
| CONF-29166 | Unable to delete blog version | | May 02, 2013 | Aug 04, 2015 | | Xavier Sanchez [Atlassian] | Abdoulaye Kindy Diallo [Atlassian] | | RESOLVED | Fixed |
| CONF-28544 | IE10 and IE11 could only print Confluence into one page, if using the Documentation Theme | | Mar 18, 2013 | Jul 21, 2015 | | Denise Unterwurzacher [Atlassian] | Hanis Suhailah [Atlassian] | | RESOLVED | Fixed |
| CONF-38555 | Stop Watching from email notification lead to 'Your request could | | Jul 21, 2015 | Jul 21, 2015 | | Unassigned | Jing Hwa | | RESOLVED | Fixed |

Using the JIRA Issues macro, you can:

- Display a table of JIRA issues on your page, based on the results of a search using JIRA Query Language (JQL).
- Display a table of JIRA issues onto your page, using a JIRA URL.
- Display a single issue from the JIRA site, or a subset of selected issues from your JIRA search results.
- Display a count of issues from the JIRA site.
- Create new issues in JIRA and display the issues on your page without leaving Confluence.

It’s a powerful macro that brings JIRA into your Confluence pages. Below, you can see how to configure the macro and find the issues you want to bring through from JIRA.

You can search for JIRA issues using JQL, create new JIRA issues, or view your most recently viewed JIRA issues.




**Pro tip:** Display issues using JIRA Query Language (JQL)

You can use the macro to display a table of JIRA issues on your page, based on the results of a search using JIRA Query Language (JQL).

JQL is a simple query language, similar to SQL, which works in JIRA. A basic JQL query consists of a field, followed by an operator (such as = or >), followed by one or more values or functions.

Examples:

- The following query will find all issues in the 'TEST' project:

```
project = "TEST"
```

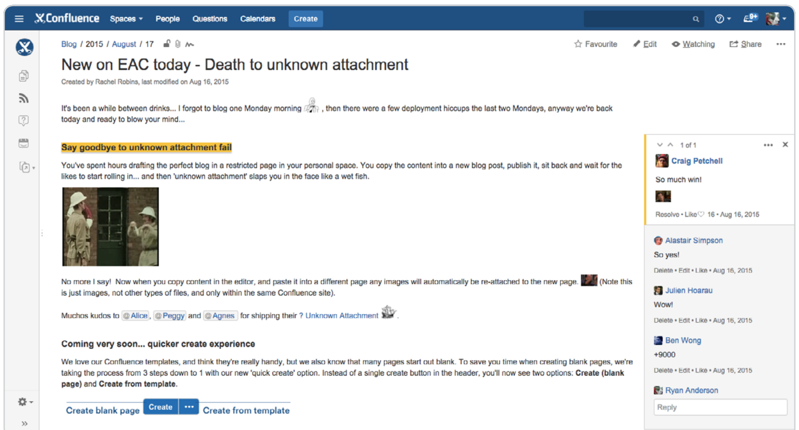
- The following query will find all issues in the 'documentation' component of the 'CONF' project:

```
project = CONF and component = documentation
```

For more information about JQL syntax, see the JIRA documentation: Advanced Searching.

Many of our release blogs are pretty simple. The *JIRA Issues macro* is a great way to quickly whip up an update on the latest product release. But often we write release blogs that are plain, old fun too—packed with gifs, funny videos, HipChat emoticons, or rambling banter. Below is an example of a blog post that one of

Confluence's technical writer's published to notify Atlassian's about some new features available to doggood on our Confluence instance that we call 'EAC'.

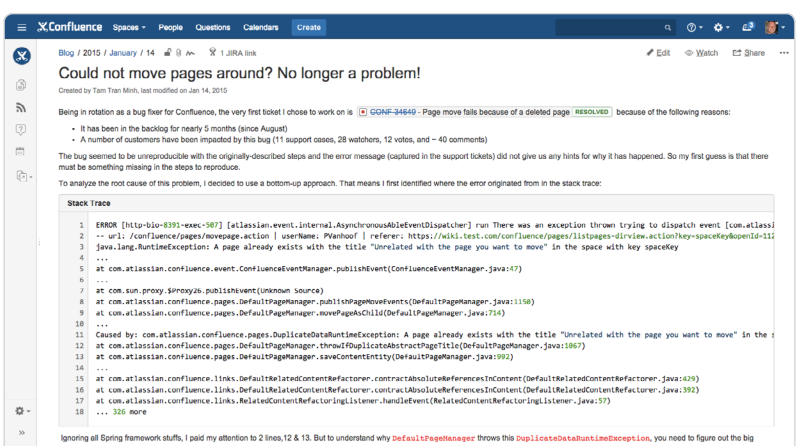


Keeping your blogs fun and light-hearted is a great way to spur engagement with your team or organization. You can see all of the supportive comments inline on the right.

## Blog 2: Sharing knowledge with the team

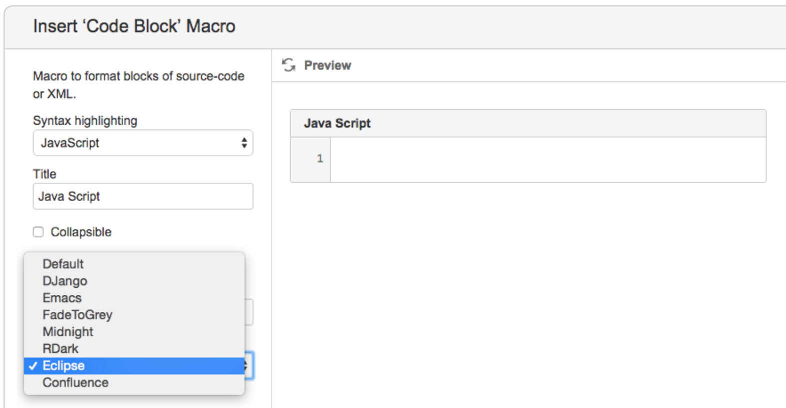
The Atlassian development teams regularly blog to get help with tough technical problems or to share how they've conquered them. One of the best ways to write about code in Confluence is using the Code Block macro.

The *Code Block macro* allows you to display source code in your page with the appropriate syntax highlighting. You can choose from a variety of different syntaxes and themes to show your code in Confluence.



To add the macro to your page, click the **Insert menu** in the editor toolbar and select '**Other macros**'. This will launch the macro browser where you can search for and select 'Code Block'. Once you do, you can configure the various fields:

Here's an example of a blog post from Atlassian's intranet that highlights how we use the *Code Block macro*.




> **Pro tip:** Speed up macro entry with autocomplete
> Type { and the beginning of the macro name, to see a list of suggested macros.

## Blog 3: New hire introductions

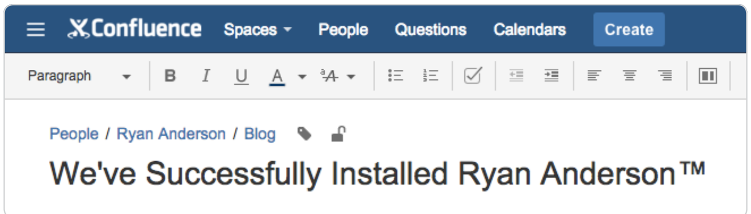
It's important to get new hires started on the right foot. At Atlassian, every employee writes an introductory blog post in Confluence. It's the perfect way for new folks to introduce themselves to the entire organization. Atlassians can welcome new hires by adding comments. It's a very positive first interaction with Confluence and the organization at large.

There are a couple key ingredients to a great intro blog post:

### A clever title

You want people to check out your introductory blog post, and the best way to draw them in is with a clever title. Here are a couple examples:

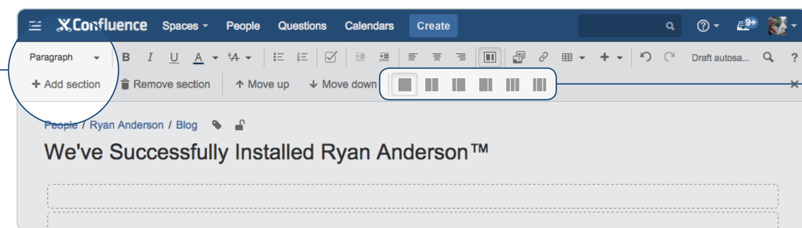
- Ryan Anderson applied to work at Atlassian and you won't believe what happened next . . .)
- We've Successfully Installed Ryan Anderson™
- New Character Created: Ryan Anderson




## Organize your blog post with page layouts

Use Page layouts to organize your blog post into sections and columns. A good tip to working with Page Layouts is to click the '**Add Section**' button a few times to give yourself some blocks to work with. Then as you build your page you can choose the various column arrangements. You can always switch them later.

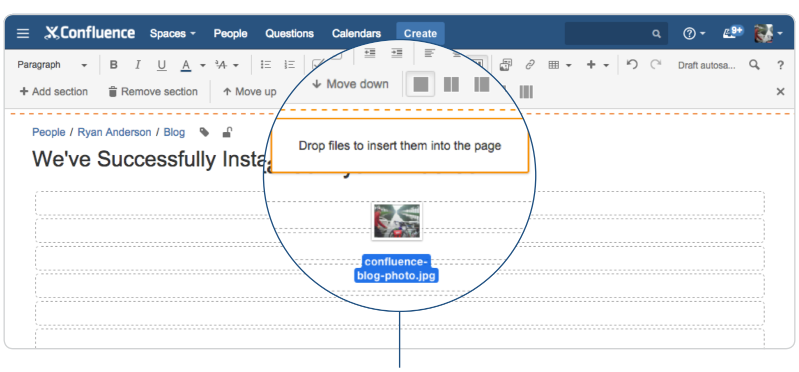
Click this button a few times when you start your page to begin organizing your content.




Click this button a few times when you start your page to begin organizing your content.

## Pictures . . . lots of pictures

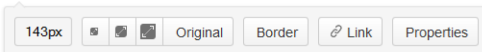
The best way to make a great first impression is to wow people with cool pictures of your travels, favorite foods, hobbies, or your family. In Confluence, adding pictures is as easy as drag and drop, so don't be shy about showing off. Just drag a picture (or pictures) from your hard-drive and drop them into the Confluence editor. Confluence will attach the images to the page and place them inline. You can then click and drag the images wherever you want on the page.



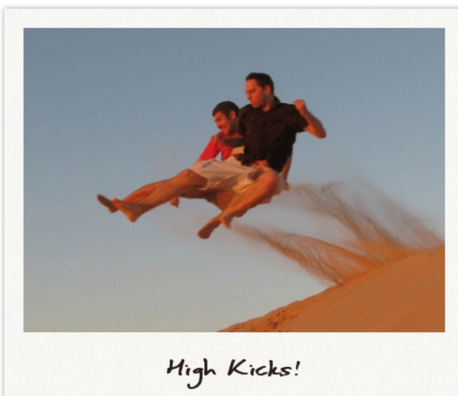

Drag and drop any image or file over the Confluence editor to drop it inline on your page.

**Pro tip:** Give your pictures some style

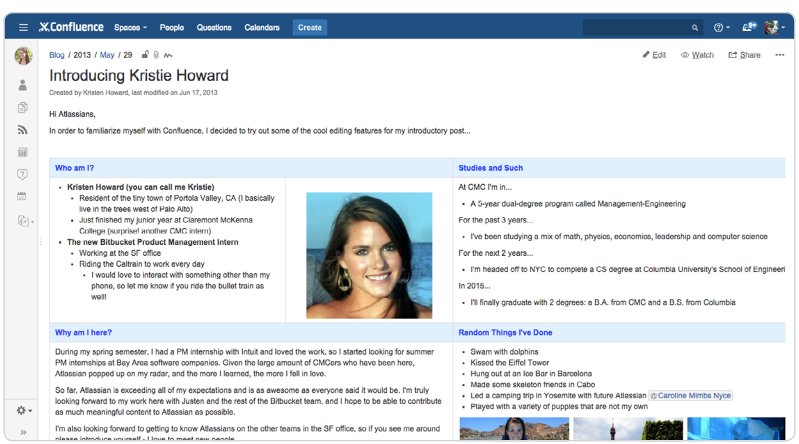
When editing the page, select the image to show the image properties panel. The panel allows you to change the appearance of an image—set the display size, add a border and effects and link the image to other pages.




By clicking the '**Properties**' button, you can give your image an effect, such as the Instant Camera, drop shadow, or curved shadow effect.




When you're all done, you should have a beautiful blog post you can be excited to share with your team. It might look something like this:



## At a glance: what did you just learn?

Blog posts are a perfect way to make announcements, offer status reports, or share timely information. Take what you've learned in this article to your team and start building a blogging culture that helps unite your development team and your greater organization. Here's a list of what we've covered:

- **JIRA Issues macro:** Bring your JIRA issues into your Confluence pages.
- **Code Block macro:** Display your source code in your Confluence pages.
- **Page Layouts:** Organize your blog post with different column layouts.
- **Adding Images:** Drag-and-drop is the fastest way to insert and image or file into a Confluence page.
- **Image effects:** Add effects to your images to give them some more color.

Chapter 8

# How to document releases and share release notes

**By Jamey Austin**

Documenting your software releases is an integral part of a software development team's job, and it's almost as important as shipping itself. Why?

- **Increased visibility:** Documenting your releases offers a window into the work your team is doing and how it's progressing toward project milestones.
- **Continous improvement:** Agile teams seek to improve every release. Having a regular cadence of release reports allows teams to see what they've done, and how they can improve.
- **Historical reference:** Always go back and see what was done in specific releases.
- **Better communication:** Clearly report to support and ops teams what was in the release and provide issue links for more detail.

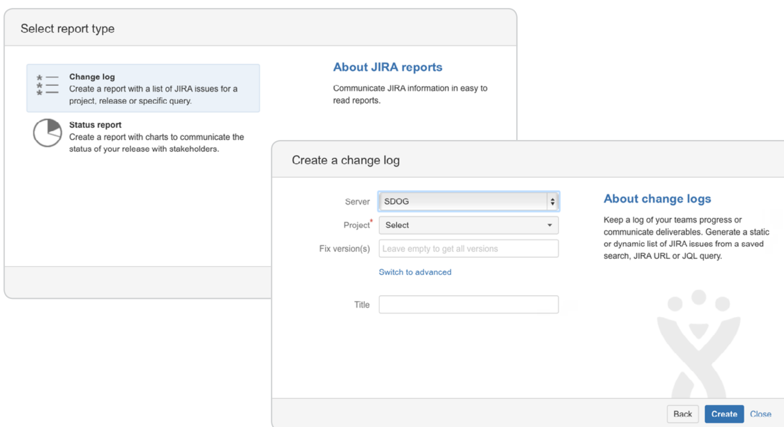
But here's a guess: this is probably one of your team's least favorite things to do. Well, we've got some good news: it doesn't need to be an annoying chore or a giant time suck.

If you track and report most of your work in JIRA, the integration between Confluence and JIRA makes it easy to insert your release results straight from JIRA into a Confluence page, where you can add more color and detail, publish internally, and also make them publicly available.

## Create an internal release report

Start with the *JIRA Reports Blueprint* in Confluence. Select **Create from template** from the top nav (3-dot button next to Create ···) **> JIRA report > Change log > Create**

Select the **Change log report** to create an internal release report. This report type gives you a detailed summary of the JIRA issues that were completed in a given release or project. Confluence guides you through creating the report and you can clarify which release you want to report on and what to include.




Select the appropriate project. (If you have more than one JIRA instance, also select which site you want to use.) Complete the appropriate fields for the **Create a change log** screen and select **Create**. Your change log report will include the data you specified from JIRA, as well as other important page elements that you can edit and fine tune.

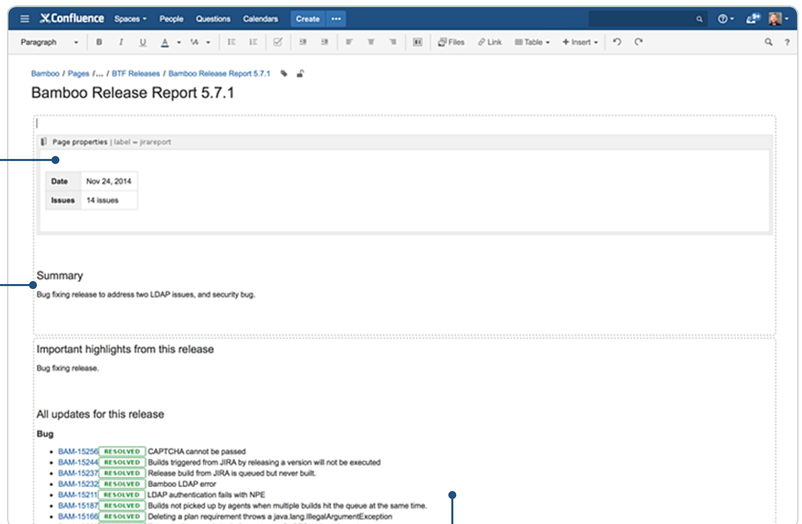
You can either create a dynamic or a static list of issues. Because release notes are a point-in-time communication, a static list is best. If you'd rather include a dynamic list, hit the **Switch to advanced** link and enter custom JQL. (See the documentation for the JIRA report blueprint for more details.)

Here's what the page looks like in edit mode.

The top of your page:

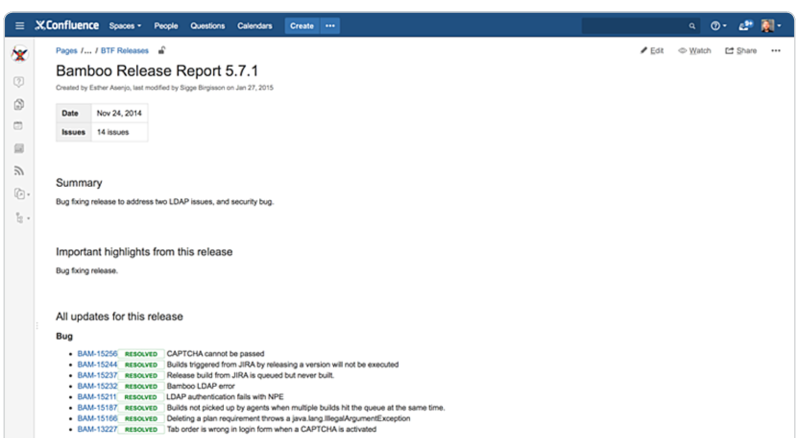
Confluence automatically pulls in the date and number of issues included in your release.

Include details about type of release report.



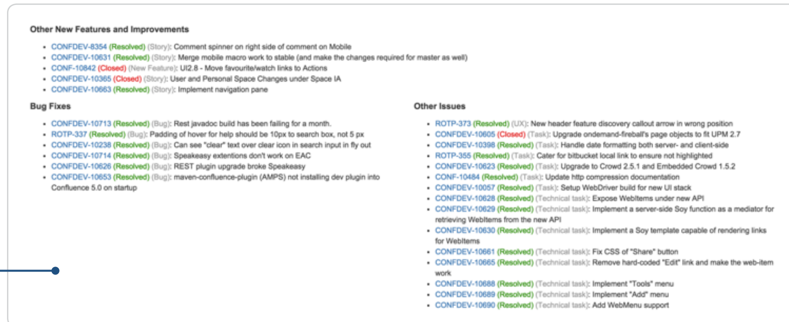

Data pulled from JIRA.

And after you've saved:




As you can see, this change log is focused on bug fixes.

Confluence automatically breaks your release down into “Bug Fixes,” “New Featured,” “Improvements,” etc.



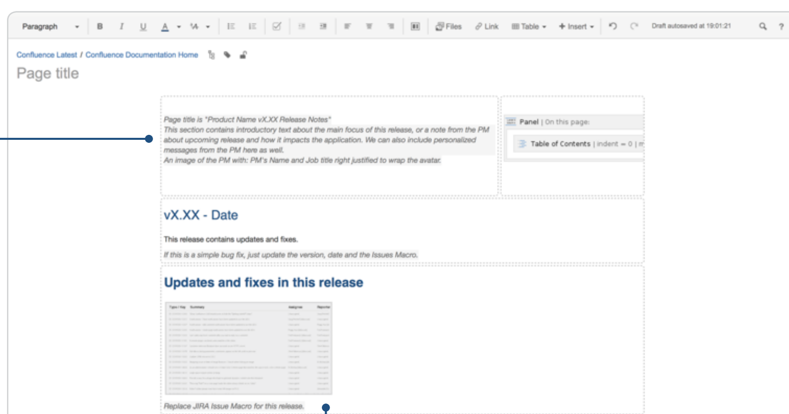

Confluence will automatically break your release down into “Bug Fixes,” “New Features,” “Improvements” etc. based on the issue type in JIRA. This automation is really helpful and eliminates the need for lots of manual editing and formatting.

## Create public facing release notes

Craft the perfect release notes for your customers again and again by creating your own release notes *template*. Here’s an example of the template we use when publishing JIRA release notes on our documentation site.

> **Templates**
> Once you’ve created a release notes page you like, Confluence lets you “templatize” that page so you can use it again and again. By standardizing your release notes with a page template, you’ll establish consistency for everyone using release notes. Learn more about Creating a template in Confluence.

“Instructional text” an be used in a template to help guide new users when creating a new page from the template.




Note use of JIRA Issue Macro.

In the template, we’ve included *instructional text* to help guide the page creator. The most important thing to include at the top is a small blurb summarizing what’s in your release, or the major theme. If it’s just a bug fix release, however, there won’t be much to add in this space. But if it's a major release, consider adding a blurb about the impact of new features, functionality, etc.

Note use of Horizontal rule from + Insert menu.


Add page enhancements.

The page includes:

**Macros:**

- Table of contents macro: Gives a brief summary of everything in your release notes.
- JIRA issues macro: Details all the issues that were resolved in the release as well as how many votes popular issues have received.

> **Pro tip:** Since you'll likely reuse certain content for your release notes, use the Excerpt + Include Page macros to include content across multiple pages instead of adding it over and over again.

**Page enhancements:**

- **Screenshots and video:** For major releases, you'll want to add screenshots and demo videos to show off new features and improvements.
- **Permissions and restrictions**: Permissions are important with release notes. Keep release notes private while you draft, and make sure to *review them thoroughly*. (*Inline comments* in Confluence are reeeeeally handy for reviewing docs.) Once they are ready for the public, you can adjust viewing restrictions. (For even more details, read our guide on using Confluence for public-facing technical documentation.)

## Edit and organize your release notes template

You can add more detail to your page templates wherever you like. You can organize your internal and public-facing release notes into what was shipped or fixed, with sections for "bug fixes" and "new features and improvements." Add macros, include images, screenshots, and video... the sky's the limit!

Just remember two things:

1. Your release notes are important to people both within your organization, and without.
2. Setting up release note templates in Confluence will help you keep both audiences informed with a minimum of hassle.

## Six content considerations for your release notes

1. **Audience:** Who will read your release notes? Think about your audience before you create them.
2. **Timing:** When will your readers need access to the release notes? Day of release? Before? Use this info to schedule release notes around deployment, to allow time for review, etc.
3. **Done:** What was completed? Separate your release notes into sections—new features, enhancements, and bug fixes. This makes it easy to see things at a glance.
4. **Edit:** If you have multiple contributors to your release notes, they might add info you don't want a reader (internal or external) to see. Edit your release notes carefully. Note: sometimes more detail, screenshots, etc. about one feature or enhancement than another will be seen as a reflection of the relative importance or complexity of that item.
5. **Facts first:** Stick to the facts. Release notes should be on point and concise.
6. **Story and message:** A storyline can help put your improvements into context. Help readers understand the changes you've made and why they're important to them.

Remember, by using a shared document that developers and product managers can update on an ongoing basis, you save yourself from having to do everything at the end. Also, your release notes *can seed other communications*. Use the content for in-app notifications, product tours, blog posts, newsletter items, or help and support material.

## At a glance: what did you just learn?

A well designed release notes page template—along with thoughtful, helpful content—will help you create release notes that not only fulfill the duty of documenting your work but something else: good, repeatable, and consistent communications that deepen your relationship with your audience.

Things to add to your release note page templates:

- **Table of Contents macro:** Create a list of heading links for quick navigation.
- **JIRA issues macro:** Detail all the issues that were resolved in the release and show issue voting.
- **Excerpt macro:** Reuse certain content for your release notes.
- **Include Page macro:** Include content across multiple pages.
- **Horizontal rule:** Divide sections of your pages by selecting from the **+ Insert** menu.
- **Permissions and restrictions:** Take control of who can view your pages with permissions and adjust viewing restrictions.

# The true power of Confluence

**By Terrence Caldwell**

The true power of Confluence is that once you're creating, sharing, and storing your information in it, it becomes the only place you need to go for all your important documents. It's the one place to find the meeting notes from the sprint planning meeting that happened last week; the design mockup for the feature coming out next month; the report from the latest release; or the blog that brags about your team kicking butt this quarter.

And the beauty is, you can use it across your entire development team, and with your extended team: product management, design, QA, technical writers, marketing, ops, and support. It centralizes all the conversations and stakeholders that support your software projects.

We hope Confluence becomes the go-to place for all your documents, the glue that connects your ideas, issues, and code to the people and conversations that bring them to life.

Happy collaborating!

**Try Confluence for free**