STRUCTURE

Jira Wins

5 Project Challenges, Solved by Someone Like You



Project and portfolio managers often get their best ideas from each other. That's what we hope you get from this book — as a Jira user, you might be fighting the same battles as our five Jira experts. Got a project growing out of control? Agile teams struggling to stay Agile? Read on for in-depth advice on struggles like these, and turn chaos into clarity.

The Structure Team

A note: Subjects may have moved on to other jobs or roles since the time these stories were written. All stories reflect the users' experience at that time.

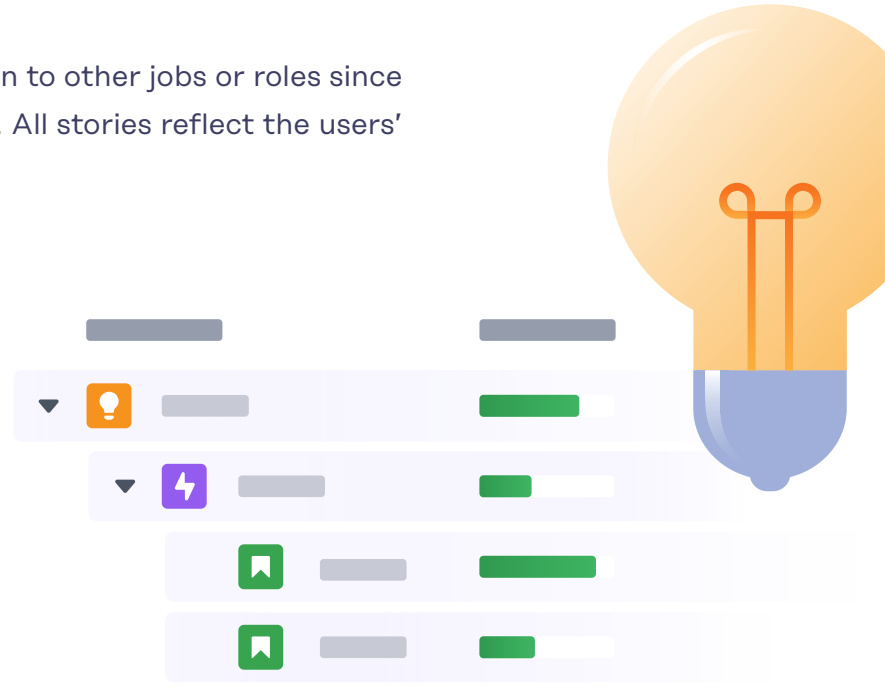


Table of Contents

CHAPTER 1	Bring clarity and order to a massive, globe-spanning project	4
CHAPTER 2	Create project predictability without stifling Agile teams	10
CHAPTER 3	Reduce requirements traceability headaches	18
CHAPTER 4	Create a single source of truth by uniting teams and processes on Jira	25
CHAPTER 5	Manage fast-growing teams as your company scales up	33

Chapter 1.

Bring clarity and order to a massive, globe-spanning project



Even experienced companies lose control of projects. Read how a mid-project reorganization brought a confusing, 75-team project back on track.

When teams can't see their progress or the context for their efforts, the work loses a sense of direction. Momentum stalls out completely.

That's the scenario Michael King was trying to fix a few years ago. King is a Program Manager for Vrbo, a vacation rental company that's part of Expedia. This story is about their efforts to migrate Australian subsidiary Stayz onto the webstack of another Expedia property, U.S. vacation rental company Homeaway.



This wasn't the company's first site migration — it had done plenty of "big bang" migrations before, where the database migration happens overnight. This time around, the combined web properties were too big, with too many bookings, listings, and moving parts to make it all happen at once. Instead, the idea was to migrate an MVP (minimum viable product), iterate, then repeat. This meant coordinating a whopping 75 teams to work on major issues like the shutdown of the legacy site, regulatory compliance, property migration, global ops, and more.

And oh yeah, these teams were on the opposite side of the world from management. Is it any wonder things got messy?

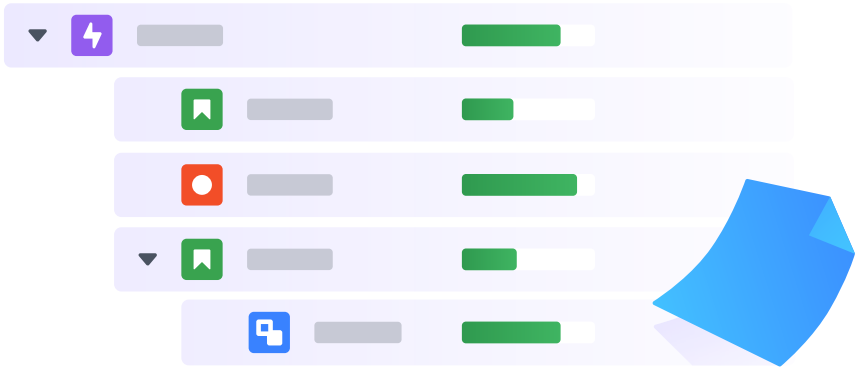
Finding the right organizational strategy

Teams were losing track of their work, and they needed a system for prioritizing tasks and mapping them to the larger project. King searched for a system to tame the chaos.

“I was really just trying to understand how work items connected to each other. Not just in terms of dependencies, but I just wanted to visualize the space. I just couldn’t understand how broad the issues were or where they belonged in individual teams or organizations — or even individual developers, when multiple people were working on very similar issues.”

His goal was to organize tasks by showing how they connect with each other. That meant breaking the bewildering pile of issues into understandable chunks so he could help the team visualize the space and see the entire scope of the project.

Using Jira in combination with Structure for Jira served those aims — especially their ability to easily group and visualize multiple epics and their child stories, so multiple epics could be viewed simultaneously even when they were dispersed across Jira projects.

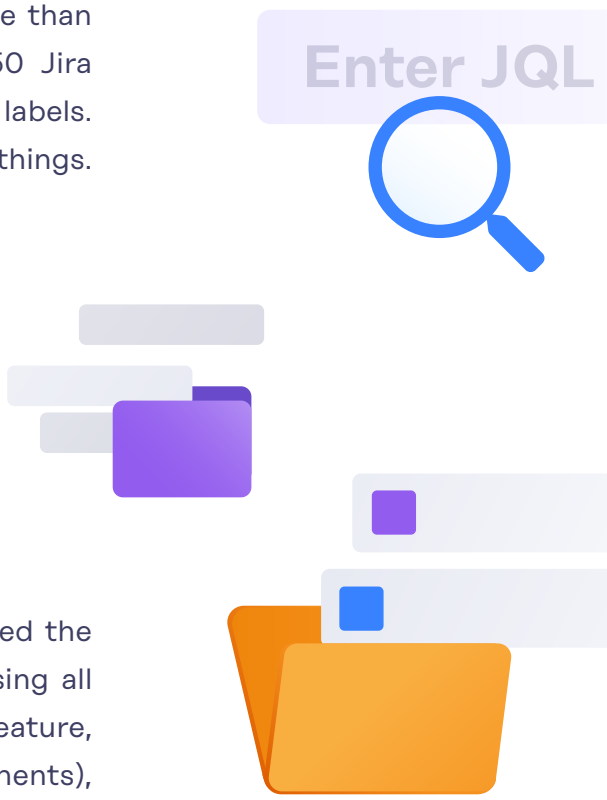


How to get it done

For a project of this scale, he created incremental delivery/release plans, called milestones. As long as everyone got their work within that bucket done, King said, management knew the project was on track. King set about sorting and organizing with Jira plus Structure.

Here's his process:

1. Find all your issues. When he joined the process, more than a thousand issues were dispersed across more than 50 Jira projects. Initially, issues were mainly identified by labels. King searched for all the relevant labels to organize things. (As an aside, King avoids labels these days — which can have misspellings or small differences that lead to confusion. Instead, he recommends using components, as they have more strict naming conventions.)



2. Sort them into milestone buckets. King built a representation of each milestone using Structure. He used the Structure Automation feature to automatically insert all of the project's related issues — current and future — into each milestone. He simply used the exact Jira Query Language (JQL) query he had been using all along to find related issues. Using the Structure Group feature, he sorted and grouped the issues by labels (later, components), and organized everything into larger milestone buckets: Milestone 1, Milestone 2, etc. To confirm teams' planned work and assigned milestones, King ran another JQL query to retrieve epics, and set about examining each epic's stories by using the Extend function to include stories under each epic, as well as sub-tasks. That provided a simple way to find child stories.

3. Find out how issues connect. Linked Jira issues identify critical paths across Vrbo's dev pods. Developers were diligently linking issues together, but had no standardized method for showing dependencies clearly. King used Structure to make the case for using "blocks/is blocked by" in cases of dependencies — standardizing on the use of "blocked" for critical path issues was enormously helpful.

4. Create a standardized method to involve teams working outside Jira. Instead of creating another system to get those official approvals of stakeholders who weren't working in Jira, King created tickets within Jira to consolidate the work there. That took effort, but it was worth it to create a single source of truth.



5. Take full advantage of communications tools. King used the app's communications functionality to keep users from getting lost amid the blizzard of issues. For example, he used the Structure.Pages extension to embed helpful Confluence pages directly under the milestone groupings in Structure, making it easier for users to reference them without ever leaving Jira. Or, if people couldn't find their way to a particular epic in the hierarchy, King used the Structure Share Perspective feature to create a URL that showed a specific milestone view that contained the epic in question.

Overall, the main achievement of all this was the visualization of a very large and very complex project. King organized all of the epics and their child stories in simple hierarchical views so that the teams could, at a glance, understand where their work belonged. By reading King's JQL queries, they could immediately know how to label their issues (or which component to use) so new issues would automatically show up in the right place.

Creating a process that lasts

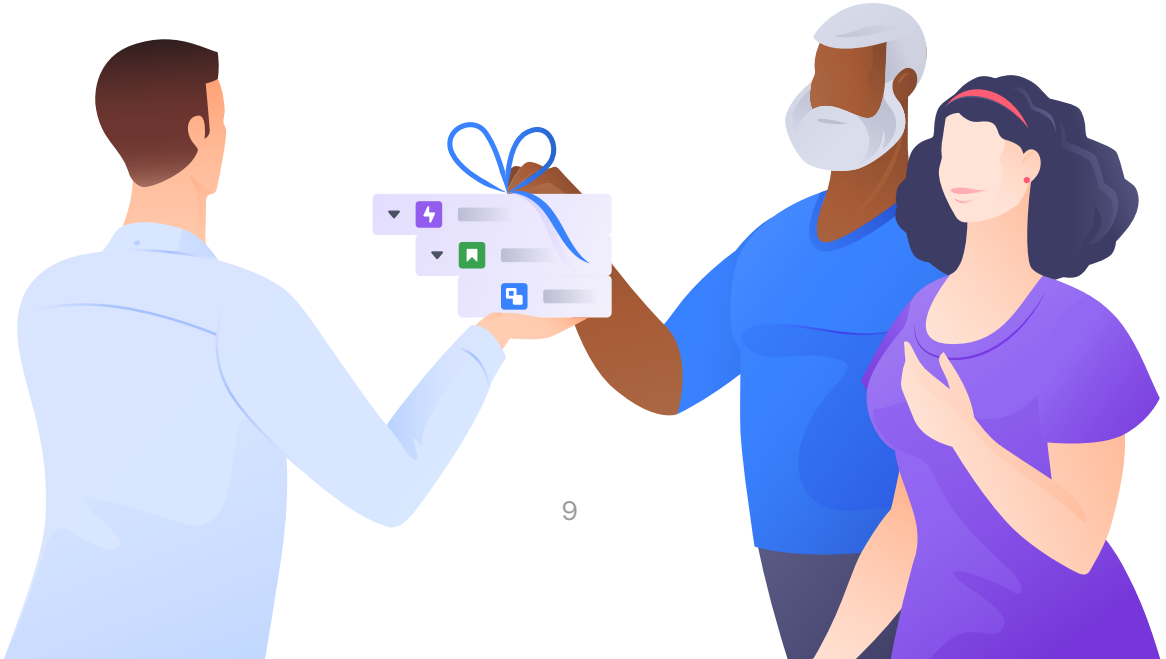
Since that project, King's Jira-powered system has been replicated for a number of web consolidations in his organization — other program managers simply adjusted it for their unique situations and put it to work.

King had searched for the right system to bring a huge, complex, international project to order, and a big part of that battle was finding a way to visualize the project in a way that made sense for everyone. Once the process clicked into place, the company took up those ideas and has run with them ever since.

See the products in action

STRUCTURE Try it free

PAGES Structure extension Try it free



Chapter 2.

Create project predictability without stifling Agile teams



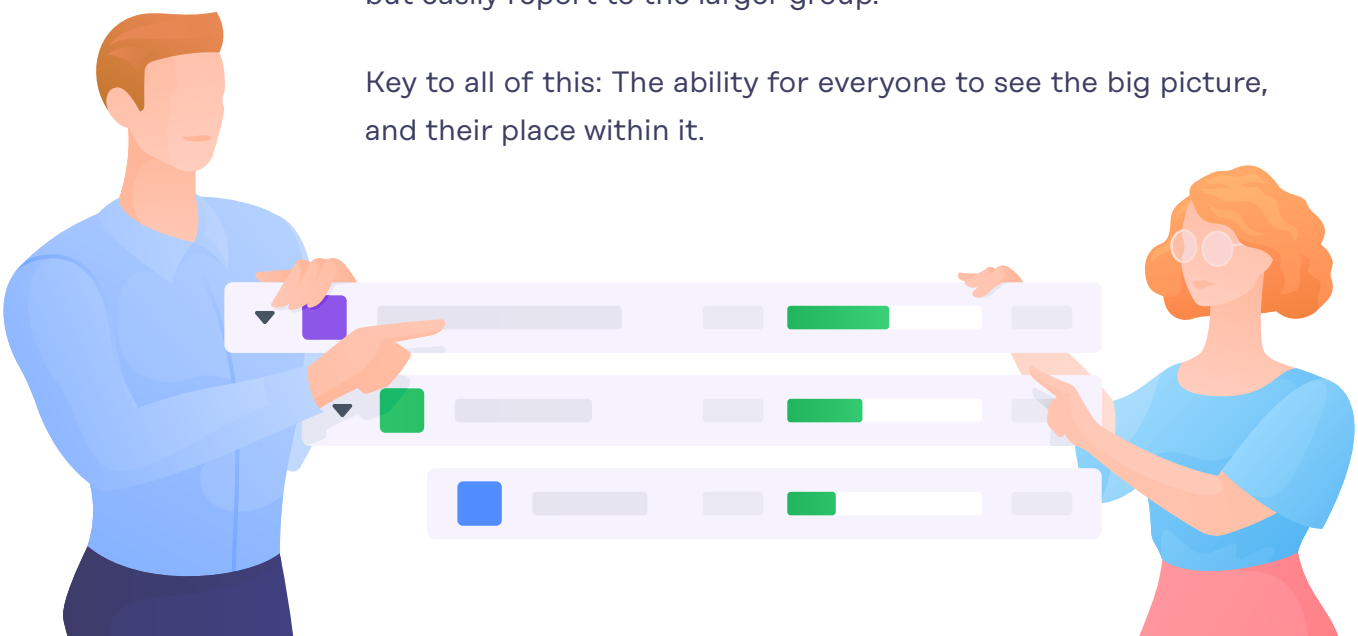
Cybersecurity provider Dragos produces and delivers solutions that protect power grids, oil and gas, and water systems, so it's vitally important that the company's product development teams deliver on schedule. Here's how they do it — without sacrificing Agility.

Like Agile teams everywhere, Dragos' dev teams want autonomy and creativity. But it's also essential that the organization deliver products and updates that their customers need — when they need them. A smart project management strategy within Jira Cloud allowed for the best of both worlds: empowered, innovative teams that coordinate within a well-orchestrated workflow.

A big picture view with Jira Cloud

Developers, quality engineers, and other contributors shouldn't have to spend massive amounts of time in update meetings or struggle to communicate their requirements to each other. They should be able to embrace Agile methods on a team level, but easily report to the larger group.

Key to all of this: The ability for everyone to see the big picture, and their place within it.



> **No matter what we do, we really need to see a holistic image of the work that's going on.**
>
> *Nathaniel Elliott,*
> *Senior Project Manager with Dragos*



In addition to Jira, he improved Dragos' system by implementing the cloud version of Structure for Jira to provide everyone with useful visualizations of the work.

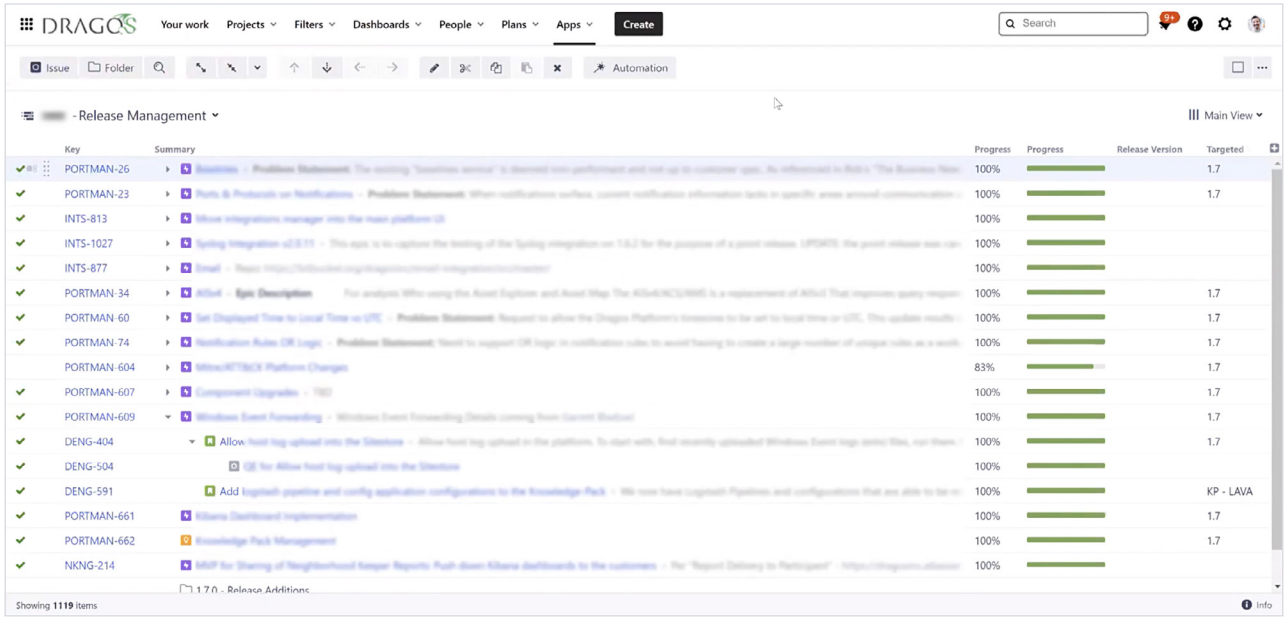
Elliot and his teams use elements of the Scaled Agile Framework (SAFe®), and the teams are aligned in projects working very closely as **component** and **feature** teams with other groups in the organization. Elliot aimed to make sure all the teams could see where Dragos was with its releases, targets, and story points.

A boost from Jira Query Language for better release management

With Structure, teams build hierarchies of Jira issues, called "structures," and populate them using the app's automation feature. Dragos uses Atlassian's Jira Query Language (JQL) within Structure to create customized release management structures where they have all of their epics broken down into features, user stories, and bugs. They have further structures for bug management, one for governance, and another called release close-out, while their front-end and feature teams have their own structures.

Thanks to JQL and the Structure Automation Generators, these structures always reflect the current state of all the relevant Jira issues.

For example, Elliott created this view so users across the organization can see overall progress without having to gather up status updates from within individual projects.



And Elliott uses Jira plus Structure to manage his own backlog of work, too. He's able to collate all of the user stories and tasks that are assigned to him, from project management tasks to reports for KPIs to the weekly readouts he has to complete. This allows him and his manager to view his list of tasks, regardless of what project they're in, so they can help him prioritize his list on the fly.

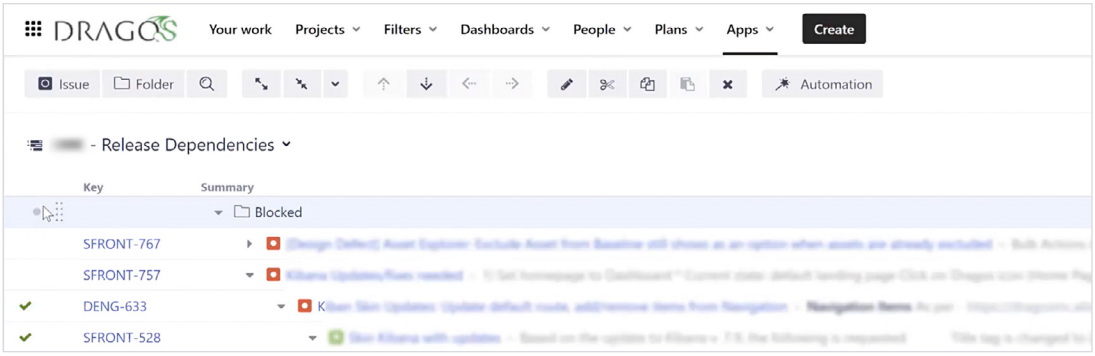
Using structures of an individual's tasks can help every person understand each other's workload and priorities, and improve coordination across the organization.

Dealing with dependency management

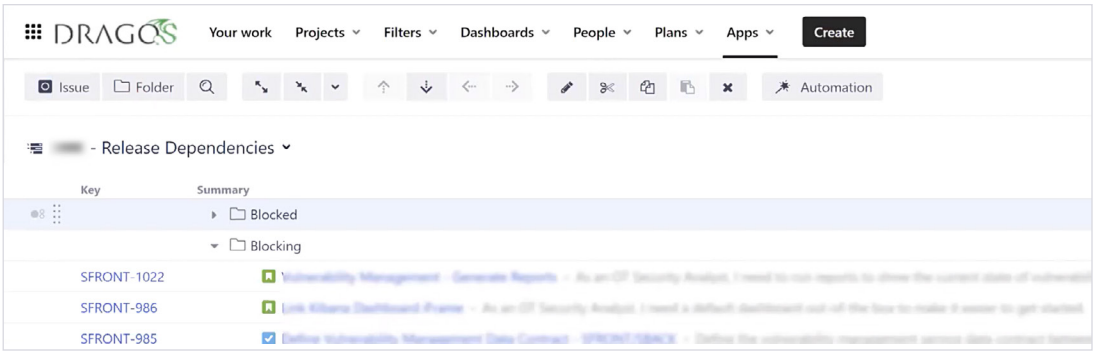
Elliott highlights the importance of dependency management, because dependencies and ad hoc requests can lead to late delivery — a major concern for Dragos. He says, “Dependency management becomes so crucial very, very quickly because I need to see what items are blocking, or are being blocked. What are the story point counts? What sprint are they going to be done in?”

Dragos uses a blocked/blocking structure. They built two release dependency folders with some very simple JQL, one for blocked tickets and one for any tickets that are blocking others. Being able to see this clearly provides them with opportunities to talk about these tickets, ask when they’re going to be resolved, and to put them into sprints ahead of time.

BLOCKED



BLOCKING

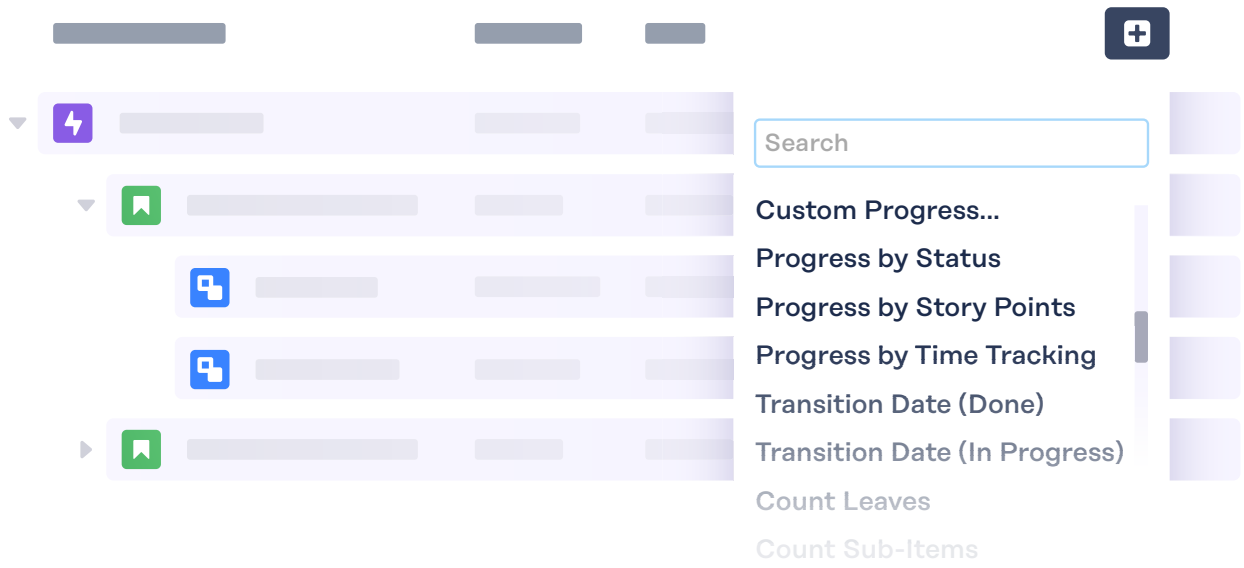


Visibility and predictability with Jira Cloud

Using Jira plus Structure allows users to create a variety of informational columns with details on almost any Jira field. That way, you can set up different views with different information, tailored to different audiences.

In another example of improved visibility, Elliott has created structures to show all the bugs they're fixing in any given Fix Version, so that they can orchestrate double and triple checks without exporting everything into Excel. The goal is to be able to push a button to say that something has been released and easily see all of the things that are in it. Elliot points out that these capabilities also make it easier to produce release notes, too.

He adds, "We always make sure views are saved for the different structures we set up so that no one needs to manually talk through updates on tickets, but instead can use out-of-the-box Jira functionality together with Structure to say, 'what is the roll-up of these story points?'"



Clarity and context for better decision-making

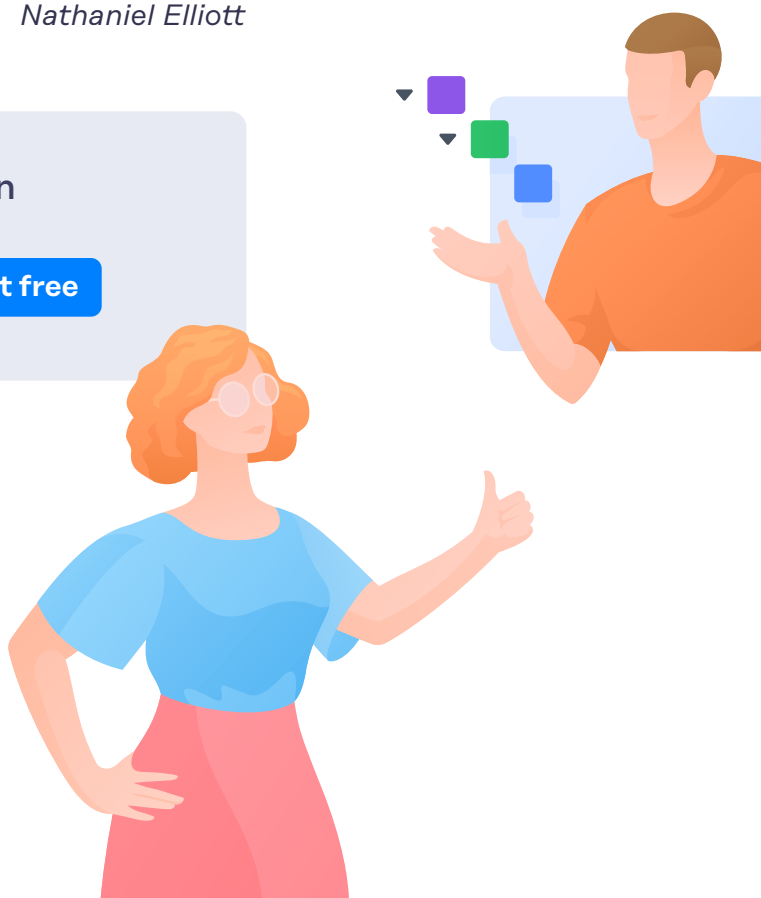
Overall, Jira and Structure have given Dragos' project managers a full hierarchical view of their projects and improved capacity to predict how things are going to go, and adjust accordingly. They have a better understanding of risks and where certain items might fall out as ad hoc requests come in. And they have full visibility of all their story points, what and when releases are going to occur, and what iteration and sprint their tickets are going to be delivered in.

> **These insights allow us to really aim towards data-driven decisions. It also helps us make sure every person doesn't need to be in every meeting.**

Nathaniel Elliott

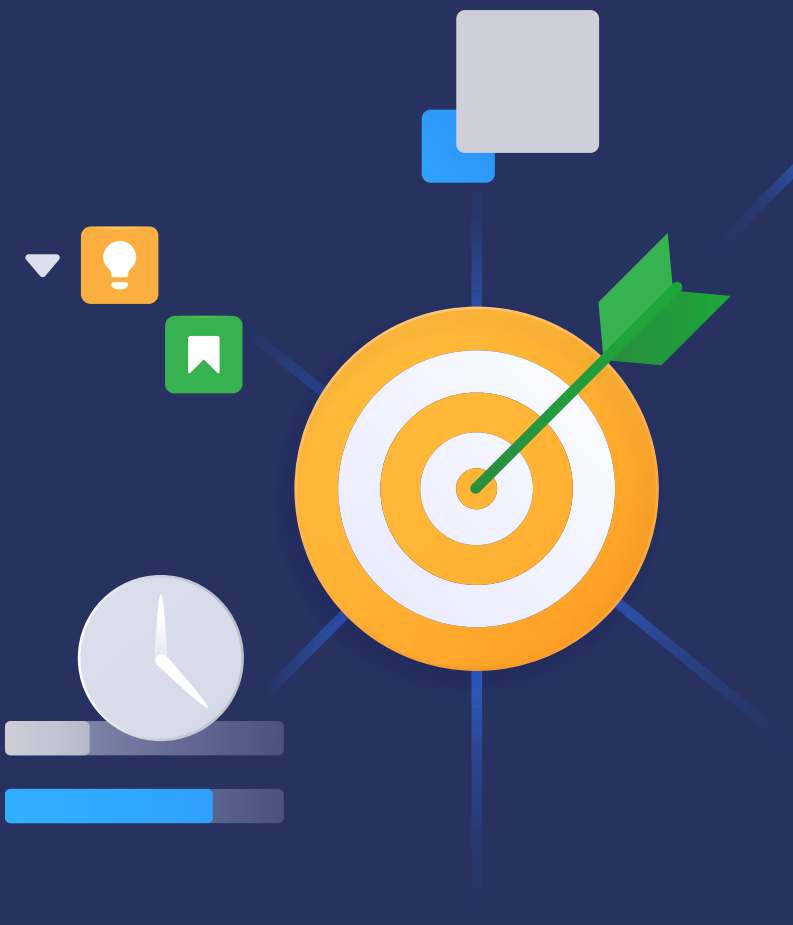
See the product in action

STRUCTURE Try it free



Chapter 3.

Reduce requirements traceability headaches



Big projects often come with overwhelmingly intricate testing requirements. Here's how one company untangled their tests.

Requirements traceability has to be one of the toughest challenges of project management. The sheer amount of information can be overwhelming: What is each component's status? Has it undergone all its assigned tests? How did it perform against the required specifications? Are those requirements linked to their components?



It's all very hard to visualize and manage without devouring hours of time to get your head around the details.

Amdocs wrestled with this problem.

A lot of functionality was not being tested.

Erez Marcovich,
a Functional Architect who is responsible
for Amdocs' requirements traceability

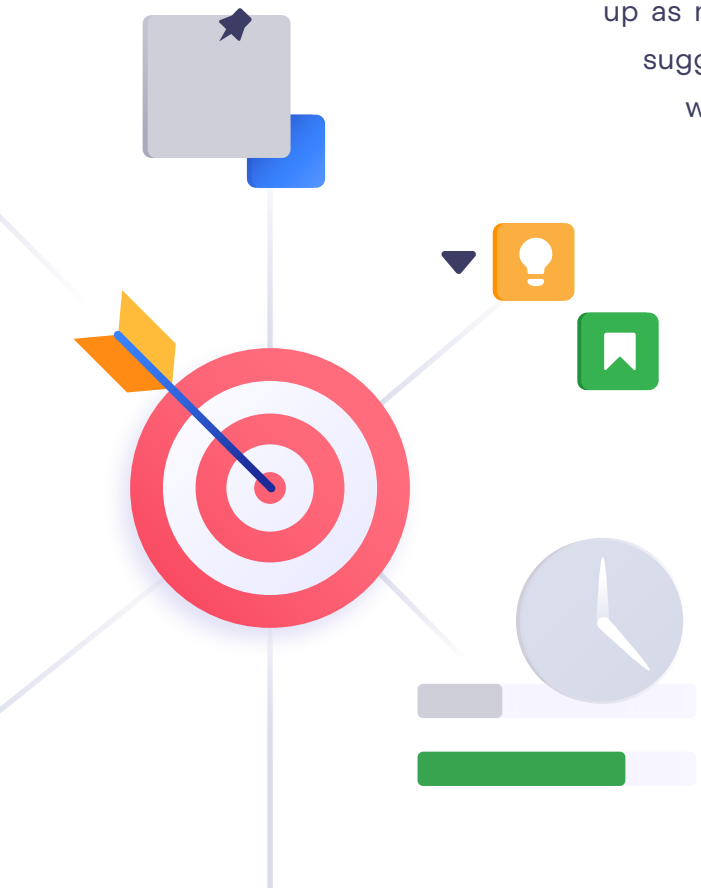
If teams can't easily see the linkages between requirements and tests, new releases could slip through without being fully tested. In those cases, the only way to know something went wrong would be when customers complained.

Their goal was to find a platform that easily linked the development requirements to the actual testing. With a clear, big-picture view, teams can understand their progress and work more efficiently. To create that system, Marcovich and Amdocs teamed up with Methoda, an Atlassian Solution partner with expertise in requirements traceability solutions.

Blending Atlassian ecosystem solutions for a unified requirements traceability platform

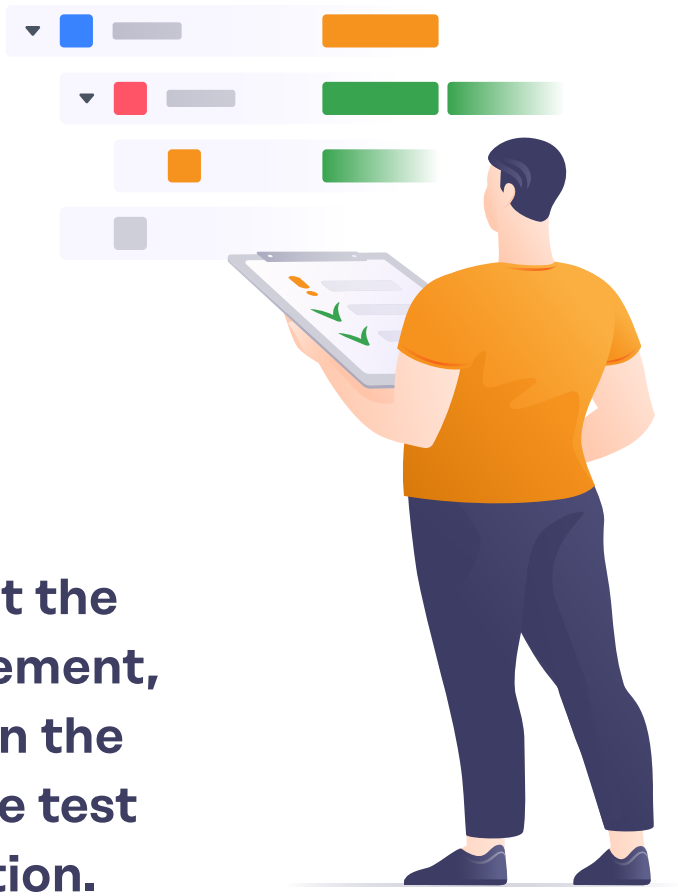
Methoda suggested a recipe to stay Agile while maintaining traceability, from requirements gathering to actual deployment.

Structure for Jira provides the basis for easy visualization. While Jira's hierarchy stops at epics, Structure allows you to build up as many levels as you like. So, for example, Methoda suggests a level above epic called "requirements," which can include its own metadata and display the aggregation of data from child issues in the hierarchy. Those requirements can be grouped under the highest level, called a "delivery."



Under the delivery level, users can see the requirements plus the status of all the requirement's children. Collectively, those indicate the status of the delivery. Structure columns are used to visualize a host of useful metadata: Progress bars to sum everything up, plus total tests, execution rate, pass rate, etc. — and these can be configured to show whatever is most useful to the customer.

Click into the requirement level, and your view drills into the specifics of that requirement. A column for "Requirement Status" then "Development Status," "Test Execution Defects," and other essential information.



> **The goal of this is to have a configurable view to present the actual, real status of a requirement, which takes into consideration the development, the flow and the test in one simple visual presentation. Structure enables us to do that pretty easily.**

Yaniv Shoshani,
CEO of Methoda

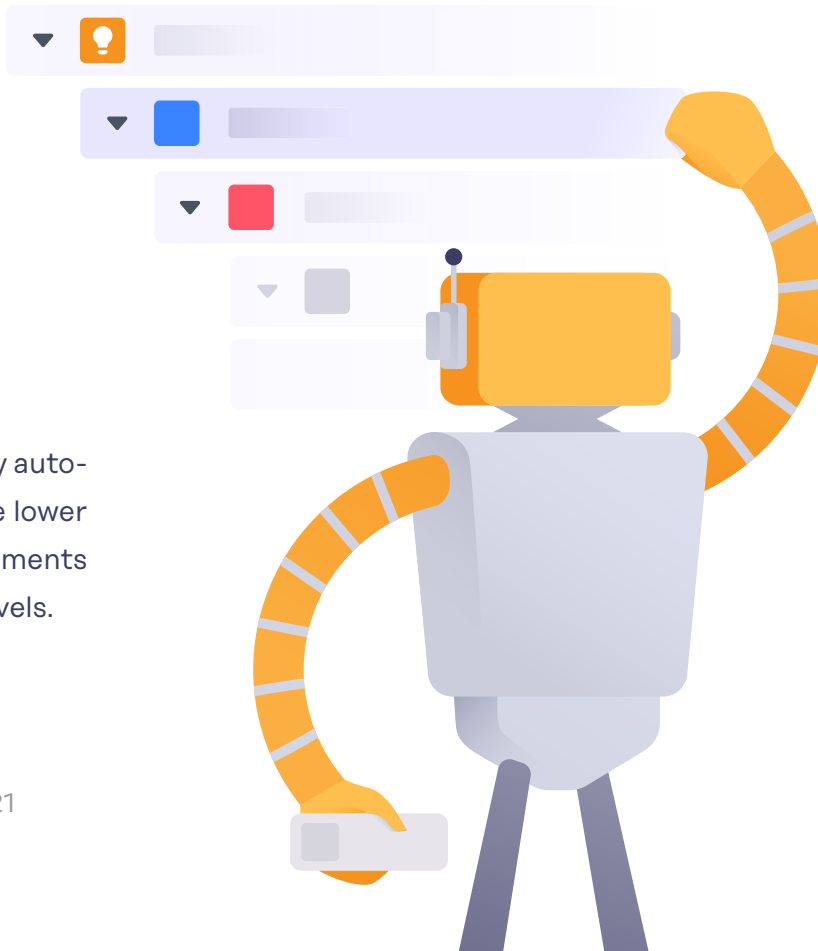
Meanwhile, the app XRay adds test management capability to Jira. Using tests and test executions linked to requirements, plus epics or user stories that are viewable in structures, the entire data flow allows users to maintain traceability and have a better understanding of the exact progress of each item.

Lastly, Methoda suggests Structure.Gantt to visualize all of the product lifecycle on a simple timeline, where one can easily view dependencies and schedule risks.

Automation that sums up statuses

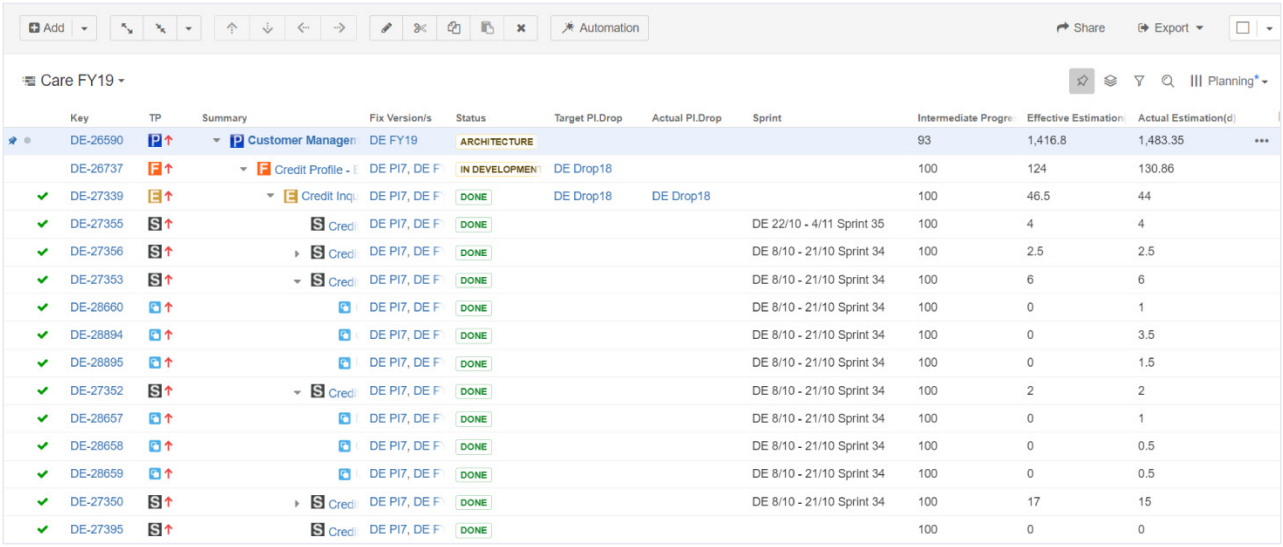
Erez and his team retooled the Methoda-suggested hierarchy to suit the Amdocs workflows. Here's what they came up with:

- Initiative
- Project
- Feature
- Epic
- Story
- Source Code
- Build Number
- Test Management
- Defects Management



With the Structure Automation feature, they automatically pull real-time information from the lower levels to show the status of all requirements on the Epic, Feature, Project, or Initiative levels.

Full Requirements Hierarchy

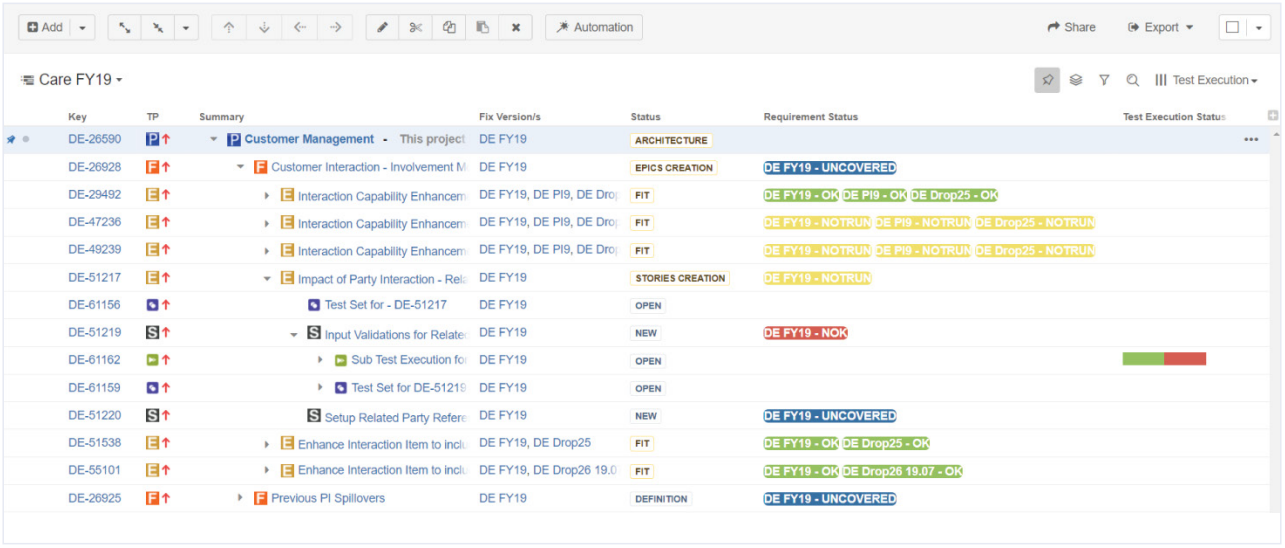


Key	TP	Summary	Fix Version/s	Status	Target PI.Drop	Actual PI.Drop	Sprint	Intermediate Progre	Effective Estimation	Actual Estimation(d)
DE-26590		Customer Manager	DE FY19	ARCHITECTURE				93	1,416.8	1,483.35
DE-26737		Credit Profile - E	DE PI7, DE F	IN DEVELOPMENT	DE Drop18			100	124	130.86
DE-27339		Credit Inqu	DE PI7, DE F	DONE	DE Drop18	DE Drop18		100	46.5	44
DE-27355		Credi	DE PI7, DE F	DONE			DE 22/10 - 4/11 Sprint 35	100	4	4
DE-27356		Credi	DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	2.5	2.5
DE-27353		Credi	DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	6	6
DE-28660			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	1
DE-28894			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	3.5
DE-28895			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	1.5
DE-27352		Credi	DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	2	2
DE-28657			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	1
DE-28658			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	0.5
DE-28659			DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	0	0.5
DE-27350		Credi	DE PI7, DE F	DONE			DE 8/10 - 21/10 Sprint 34	100	17	15
DE-27395		Credi	DE PI7, DE F	DONE				100	0	0

Amdocs also configured its visualization to include columns describing intermediate progress (again, using information from lower levels on the hierarchy), effective estimation, and actual estimation.

As Methoda suggests, Amdocs also uses an XRay extender to provide additional information in their hierarchies — primarily useful to the testing manager and release manager — to provide a broader view of progress from a number of angles. For example, the below is a view of testing coverage and execution, where you can drill down to each story's test execution and see at a glance the status of each test. You can instantly see which story is in "Not OK" status because half its tests were failing.

Testing Coverage and Execution Within the Requirement View



Amdocs instituted its new system amid some skepticism from veteran testers at the company.

> **But in a very short while, they embraced it. Now, no one wants to go back.**
>
> *Erez Marcovich*

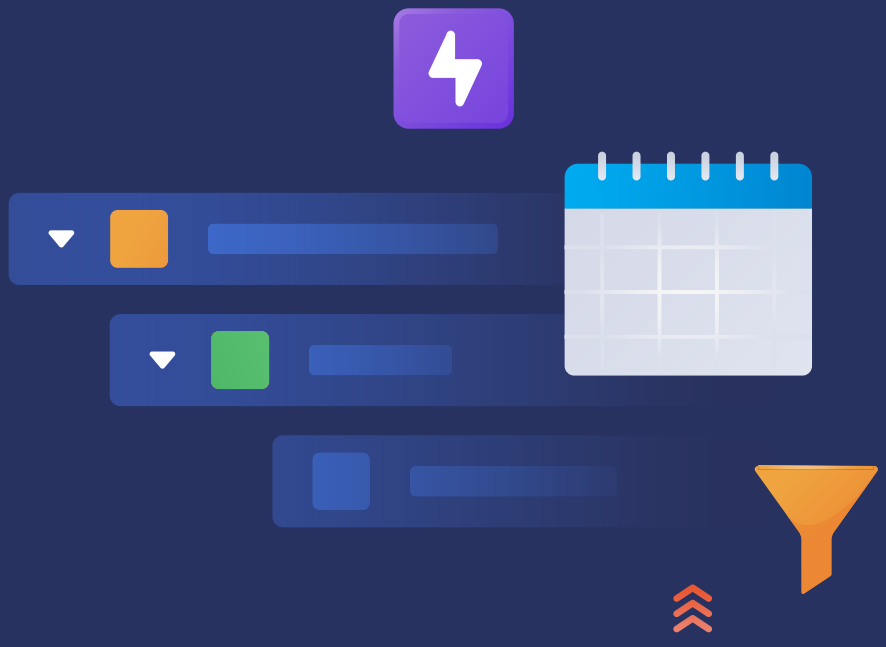
See the products in action

STRUCTURE Try it free

GANTT Structure extension Try it free

Chapter 4.

Create a single source of truth by uniting teams and processes on Jira

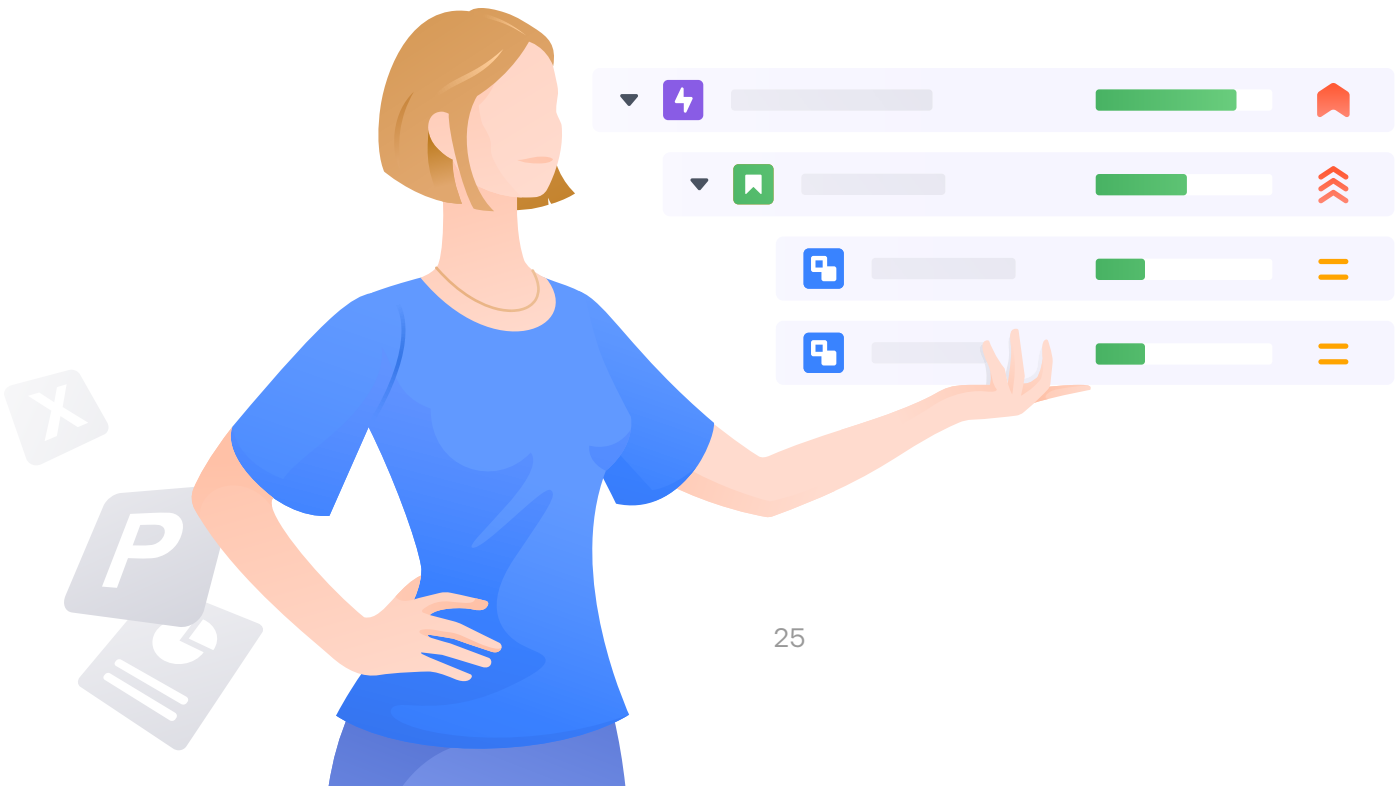


Kasia Allen, a seasoned project and portfolio management pro, likes a tidy project process. So when she found a fragmented, multi-platform workflow in her new role at an IT services and consulting company, she worked to streamline efforts within Jira.

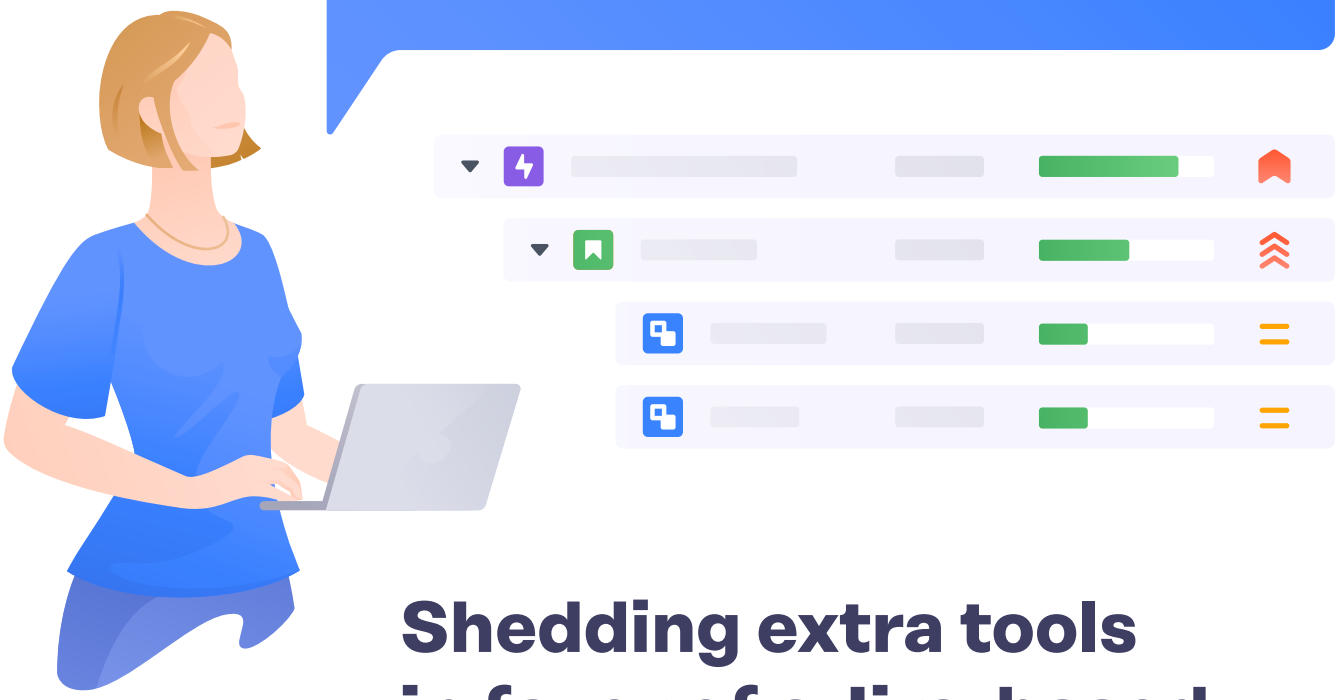
When Kasia Allen took over as head of Web Operations and PMO at her company's web marketing organization a few years ago, she found a process that incorporated different platforms — leading to way more work for everyone.

Major initiative planning took place in PowerPoint, but the actual execution would move over to Jira. When plans changed, someone would have to update the PowerPoint, then update the work in Jira — which of course meant that you couldn't be sure that everything was in sync.

What's more, the department used tools that didn't adapt to its process or priorities, forcing an organizational system that didn't match what was really happening.



> “We needed a dynamic, single source of truth that gave the senior leaders and my team access to real-time, consistent data in one location. Our solution also had to be incredibly flexible and reactive to reflect the way our teams work and constantly adapt to fluctuating priorities.”
>
> *Kasia Allen, Web Director*

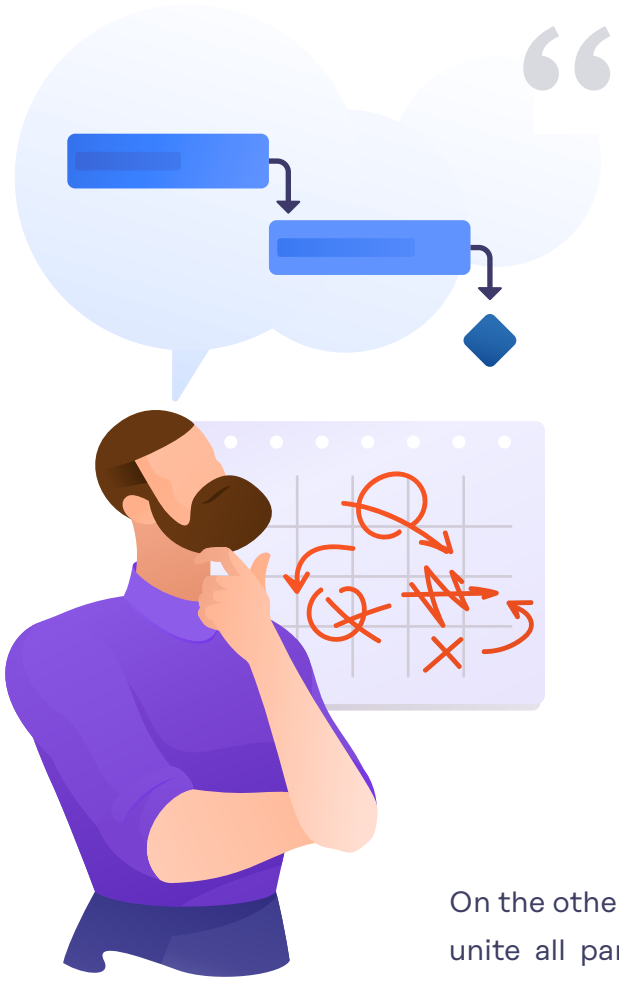


Shedding extra tools in favor of a Jira-based system

Allen’s job entailed managing ongoing processes, setting priorities for the future, and working in an Agile development environment for front-end and back-end development.

When she first took on the job, the mix of PowerPoint in SharePoint, Jira and an additional Gantt app disconnected planning from execution, leading to inefficiencies in coordination.

PowerPoint was a great presentation tool, but it was static and needed to be manually updated as soon as anything changed. And with task execution happening in Jira, it couldn't act as a single repository for all project data. Meanwhile, the company used a Jira Gantt app for planning purposes as well — but Gantt put artificial restrictions on the department's planning process. For example, this Gantt app required specific start and end dates for tasks.



> **It forced us to create pretend timelines for things — when a lot of times, a task just had to get done at some point in the quarter, with no specific date. Besides, we have ever-changing priorities. Something that is the most important thing today might be on the back burner tomorrow.**

On the other hand, Jira was a "living, breathing thing" that could unite all parts of the process — making it the single source of truth, where all data resided. Everyone could be confident that data was up to date, and it was flexible enough to better match the organization's processes. Visualization across projects was the missing piece.

A friend recommended Allen try Structure for Jira. When she called the Structure team, "I literally just described my situation and what I wanted to do and asked, 'Can you help me with this?' And the support team just walked me through it."

Organizing cross-project work in a single view

Working in Jira eliminated the planning/execution disconnect. Allen then looked for a solution that would help her plan and prioritize across Jira projects, and then organize all that information within a single view. Allen ultimately used a variety of Structure features to expand upon Jira's native capabilities.

Sprint planning with automatic updates

"This was the game-changer," Allen said. She combined sprint planning across projects within Structure and changed effort estimates from time spent to story points. Allen used Jira Components for assigning teams, so she could get summaries of story points per sprint per team.



When story points are assigned to both QA and development efforts, those Jira Component summaries are automatically updated — previously, Allen had had to use JQL queries for each team, continuously refresh them to update everything, and export to an Excel spreadsheet to analyze team capacities and make adjustments.

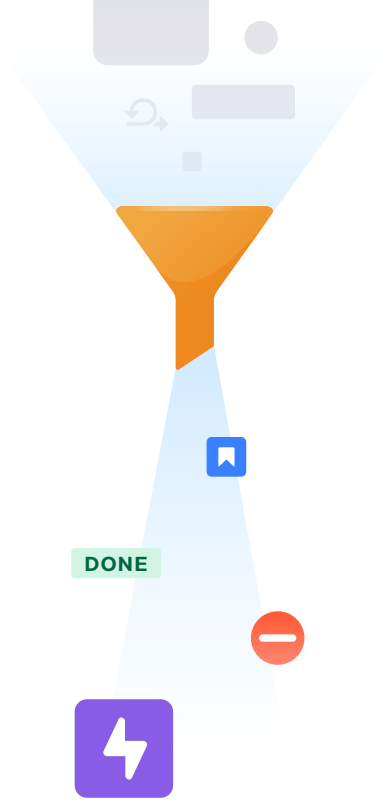
After the change, the planning could be done within Jira + Structure, where Allen could add or remove story points and watch the entire project hierarchy automatically update.

Filters to easily monitor project progress

Everyone — from a VP looking at a big-picture view, to the developer working on a Jira issue — can find the information relevant to them. The key is that everyone works off the same, most up-to-date information from a single source of truth — Jira. For example, they can see what items are assigned to a team or individual this quarter, or the state of dependent and predecessor tasks.



Allen added fields to each issue — instead of just seeing the assignee for each task, she created fields to record the name of the QA engineer, the content strategist, and the project manager related to each issue. Anyone who filters for their own issues (the "Mine" filter) can make sure they are not a blocker — a critically important point, as her workplace had teams across the globe. When they start their day, they can view their tasks with one click instead of doing a JQL query manually into Jira.



Structure Filters are highly useful for a variety of things: Allen used them to see who hadn't yet added their level of effort estimates, or to note which items were blocked, or to see which items should or shouldn't be in a particular sprint.

They proved useful for release planning as well. Allen tagged everything to a release and saw what issues were related — with one click, her team could view all the related items and have a much more productive discussion about everything related to an upcoming deployment.

Adaptable hierarchies

The new system also builds upon Jira's flexibility, allowing the organization to create an adaptable hierarchy that reflects her teams' existing processes and issue hierarchies.

For example, when she first took the job, Allen tackled a massive backlog of tickets that had been stuck in limbo — in part because they were hard to categorize under the Gantt app they were using at the time. Every issue had to have a parent epic, but that wasn't universally true across all projects in the company.

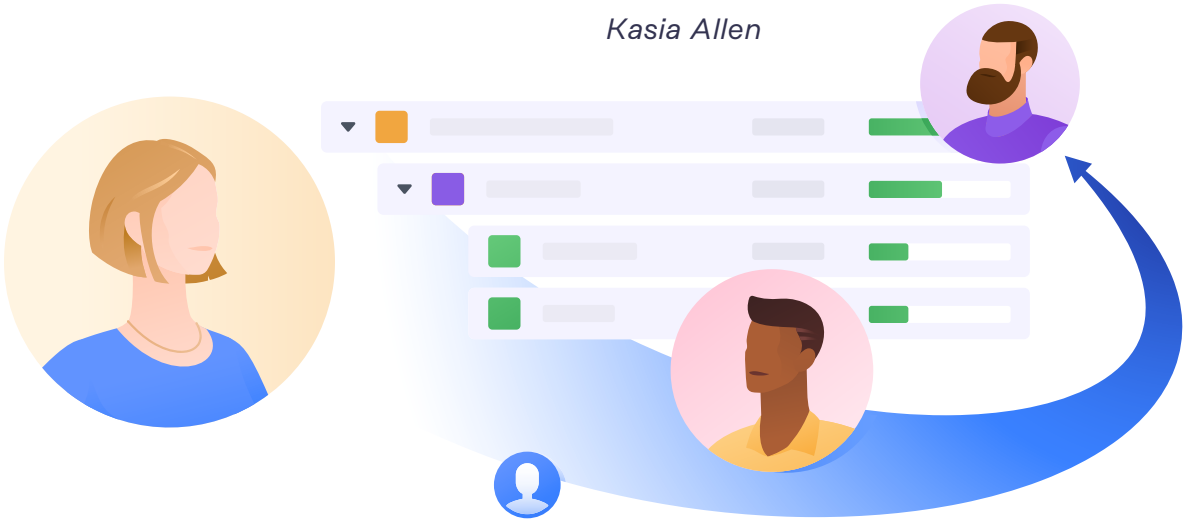
> **We don't always work in an epic world... In our world, a singular issue or bug can be a huge priority. I need to visualize it, so previously we had to create a fake epic just to add it to the chart so it shows up in the sprint — it's just way too much overhead.**
>
> *Kasia Allen*



Structure allows for those one-off issues to have their place in the hierarchy, without the need for artificial categorizations.

Altogether, the changes freed up a tremendous amount of time for Allen and her team. Trying to connect the disjointed planning and execution process took up hours every week that is now available for more strategic work. Uptake within her organizational level was also not a struggle, Allen said, as more teams willingly joined the Jira system once they saw how much easier it was to coordinate work.

> **As an uber-organized person, I have to see what's going on, and this system put me in control.**
>
> *Kasia Allen*



See the product in action

STRUCTURE Try it free

Chapter 5.

Manage fast-growing teams as your company scales up



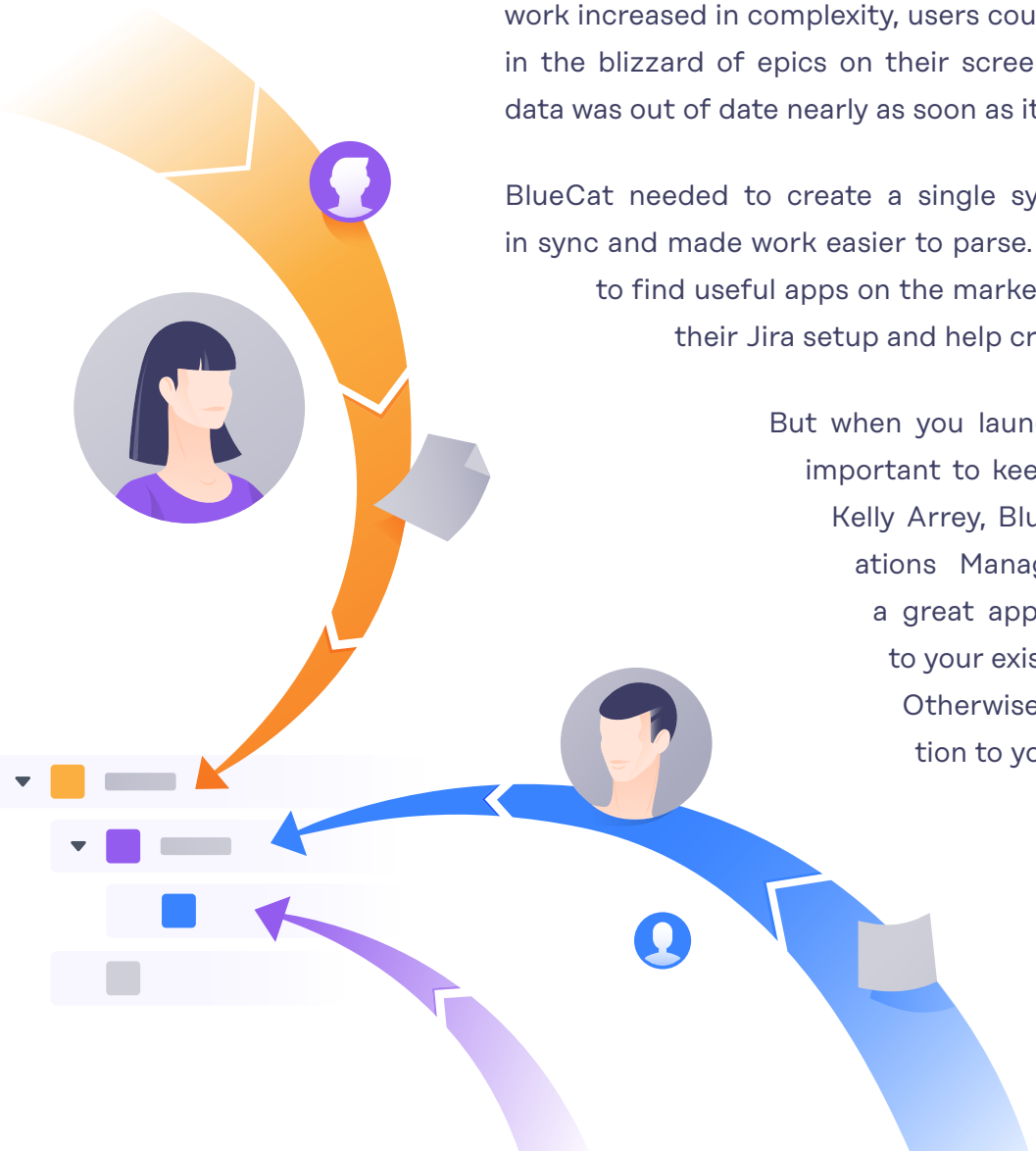
One company's simple rule:

> **Always look for project management tools that are flexible enough to suit how you already work.**
>
> *BlueCat Networks*

A few years ago, BlueCat Networks' project management process was fraying at the seams. Their tooling was a combination of spreadsheets and out-of-the-box Jira, and that had worked well back when the Toronto-based software company had only two or three teams in the mix — but now, BlueCat was growing. As its work increased in complexity, users couldn't see their work clearly in the blizzard of epics on their screens, and their spreadsheet data was out of date nearly as soon as it was exported.

BlueCat needed to create a single system that kept everyone in sync and made work easier to parse. Clearly, the next step was to find useful apps on the marketplace that could build off their Jira setup and help create a scalable system.

But when you launch a search like that, it's important to keep one thing in mind, says Kelly Arrey, BlueCat's Engineering Operations Manager. Don't just look for a great app; look for one that maps to your existing workflow and culture. Otherwise, you're just adding friction to your organization.



"If your tool maps to your process in a straightforward way, your users don't need to stick Post-it notes all over their monitors to remind them how to navigate it — at each point, the next step is clear. When it's easy to use, people will use it."

BlueCat landed on Structure for Jira as its primary way to efficiently track and understand their efforts clearly as they scaled up. Here's how the company, which had a 130-person dev team as of this writing, organizes its work in Jira with the help of Structure.



BlueCat began their Jira (and Structure) journey with the server edition of those products. Like many Atlassian on-prem customers, they recently embarked on a new journey — to Atlassian Cloud.

> **We're actually using Structure Cloud to manage our migration to Jira Cloud.**
>
> *Kelly Arrey*



Organize the work to match your team's preferences

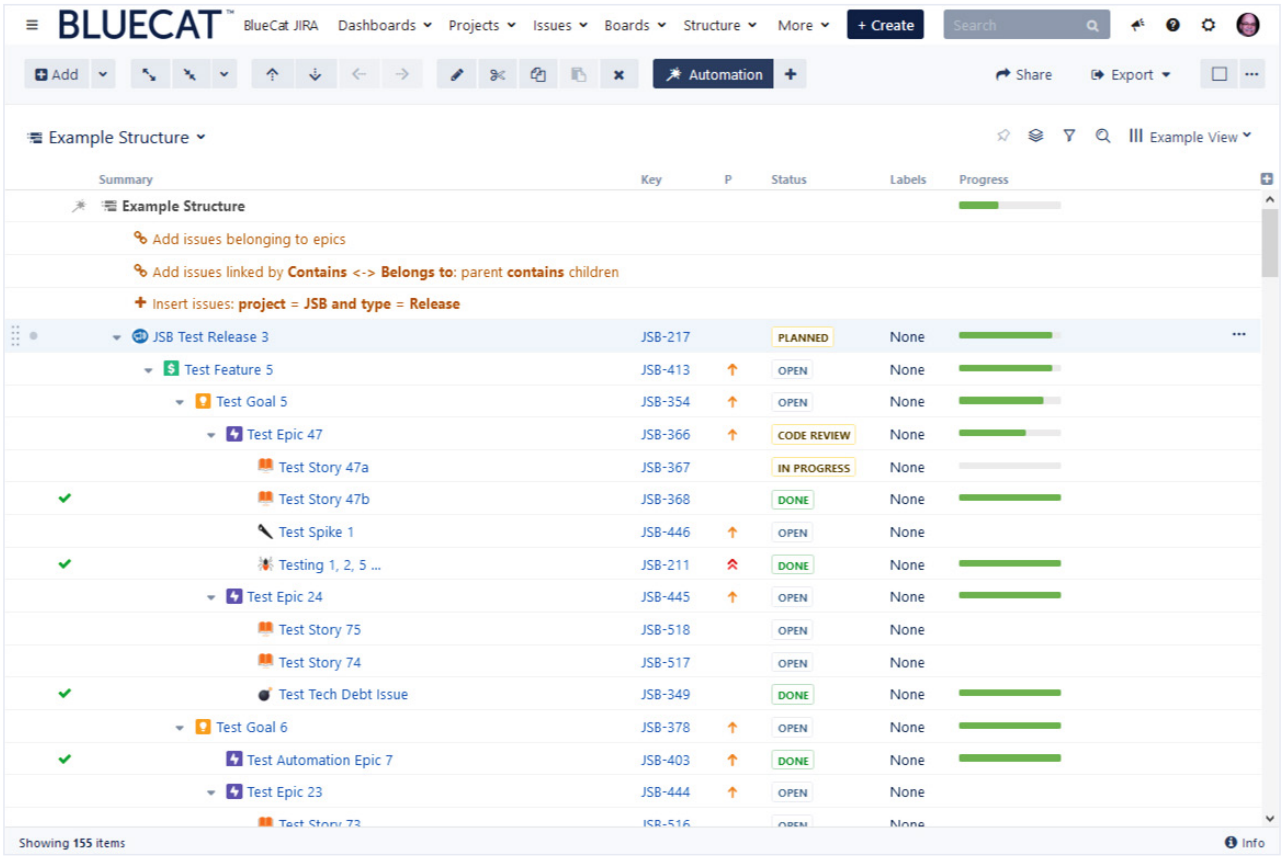
If your work isn't organized in manageable pieces, and in a way that is somewhat intuitive for your team, you'll constantly miss important data and your teams will struggle to coordinate.

BlueCat wanted to organize hierarchies that went above the standard Jira epic. Structure allows for largely open-ended construction of hierarchies, so Bluecat could organize their hierarchies to suit their preferences. Here's what they went with (see Fig. 1):



Figure 1.

BlueCat could zoom in and out of its hierarchies — drilling down to see stories, and then back up for high-level feature views. Everyone could see their work in its proper context, in as much or as little detail as they needed.



BlueCat is able to visualize their preferred issue hierarchy with a few simple rules.

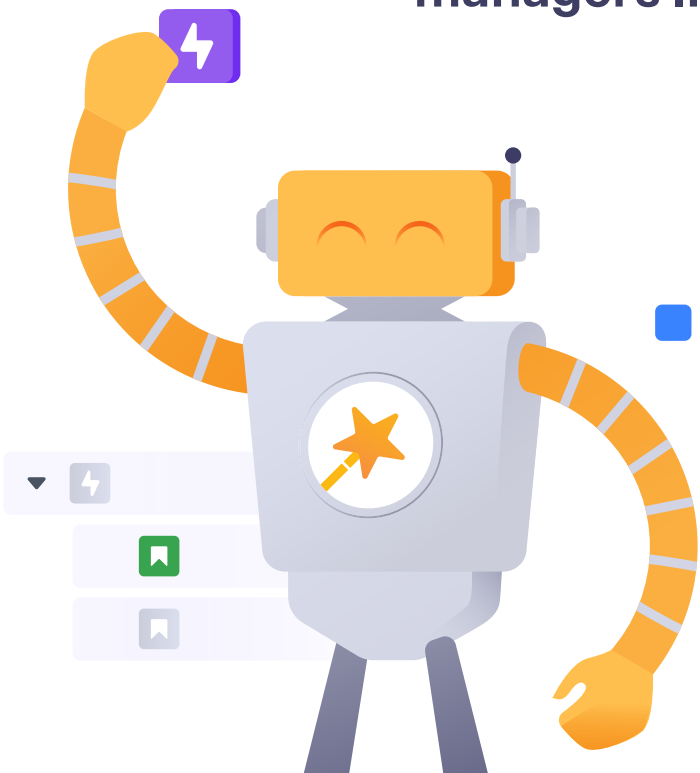
Keep key connections in place, automatically

With Structure, Arrey's team builds hierarchies of issues, called "structures," and populates them using the app's automation feature. That feature allows the users to define special rules — called generators — that become the basis for these hierarchies

BlueCat uses an Insert generator to pull in the root-level issue and they add Extend generators — which follow issue links, epic links, and sub-task links — to build out the tree from the roots.

For example, BlueCat uses a custom Jira issue link called "contains < > belongs to" to create the desired hierarchy. If someone updates the issue links, the structure updates in response. Furthermore, as new "releases," "features," "goals," etc., that fit the JQL criteria are created elsewhere in Jira, they are automatically pulled into the structure.

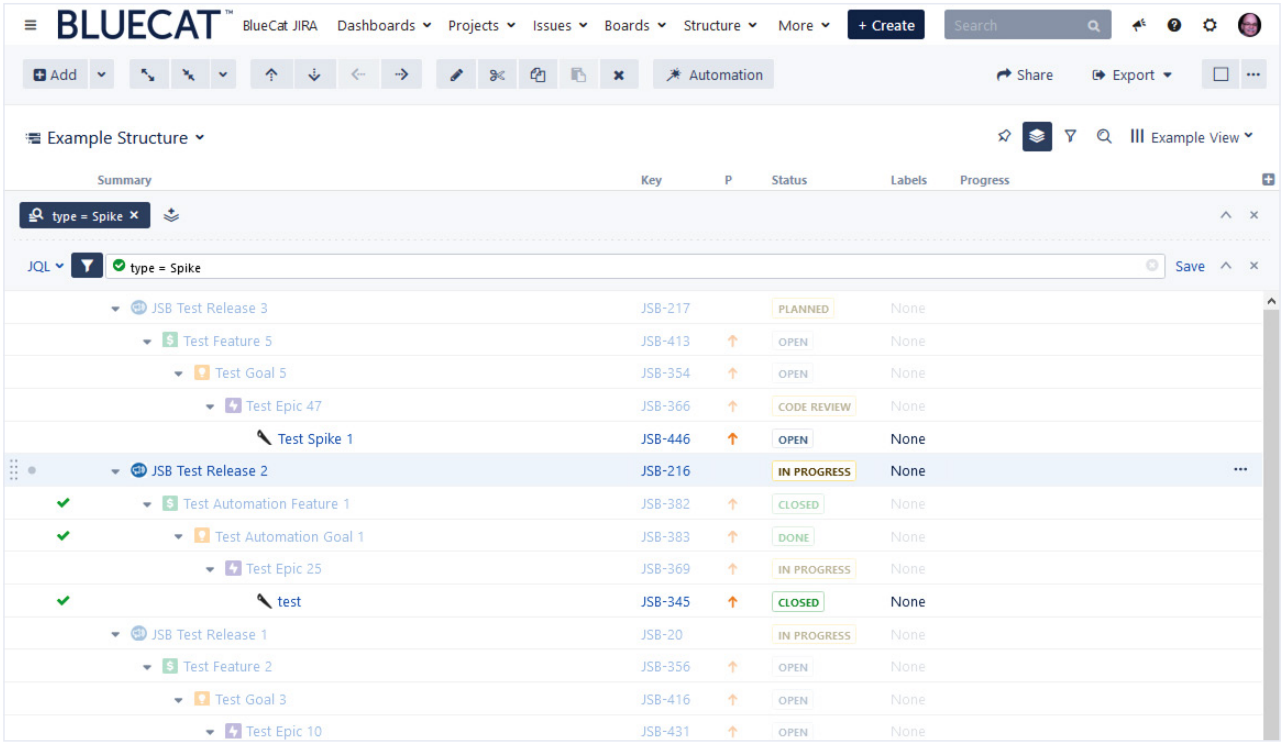
> **That's very powerful — our program managers live and die by this.**
>
> *Kelly Arrey*



De-clutter hierarchies to keep users focused on the right tasks

Individual BlueCat users can take advantage of a feature called Transformations to re-organize their hierarchies — without changing the views of other users sharing the structure — to see individual tasks in a variety of contexts that can help them ensure everything is accounted for.

These allow users to easily see the issues that matter to them without affecting anyone else's view.



BlueCat uses Transformations to narrow the focus of their structures for a particular topic or team.

BlueCat uses a few types of Transformations. Select an epic together with the pin Transformation, and you zoom in on specific parts of the structure. For example, selected epics — and their parents and children — will be highlighted while everything above and below will be grayed out or disappear.

Other Transformations create different views that allow users to see their work in whatever context they need.



Adapting as your workflow evolves

Arrey wanted to construct a suite of tools that mapped to Blue Cat's workflows and processes — and in part, that meant creating a system that could change as necessary.

> **Our development process continues to evolve. At any given time, we may update it to suit our needs.**
>
> *Kelly Arrey*

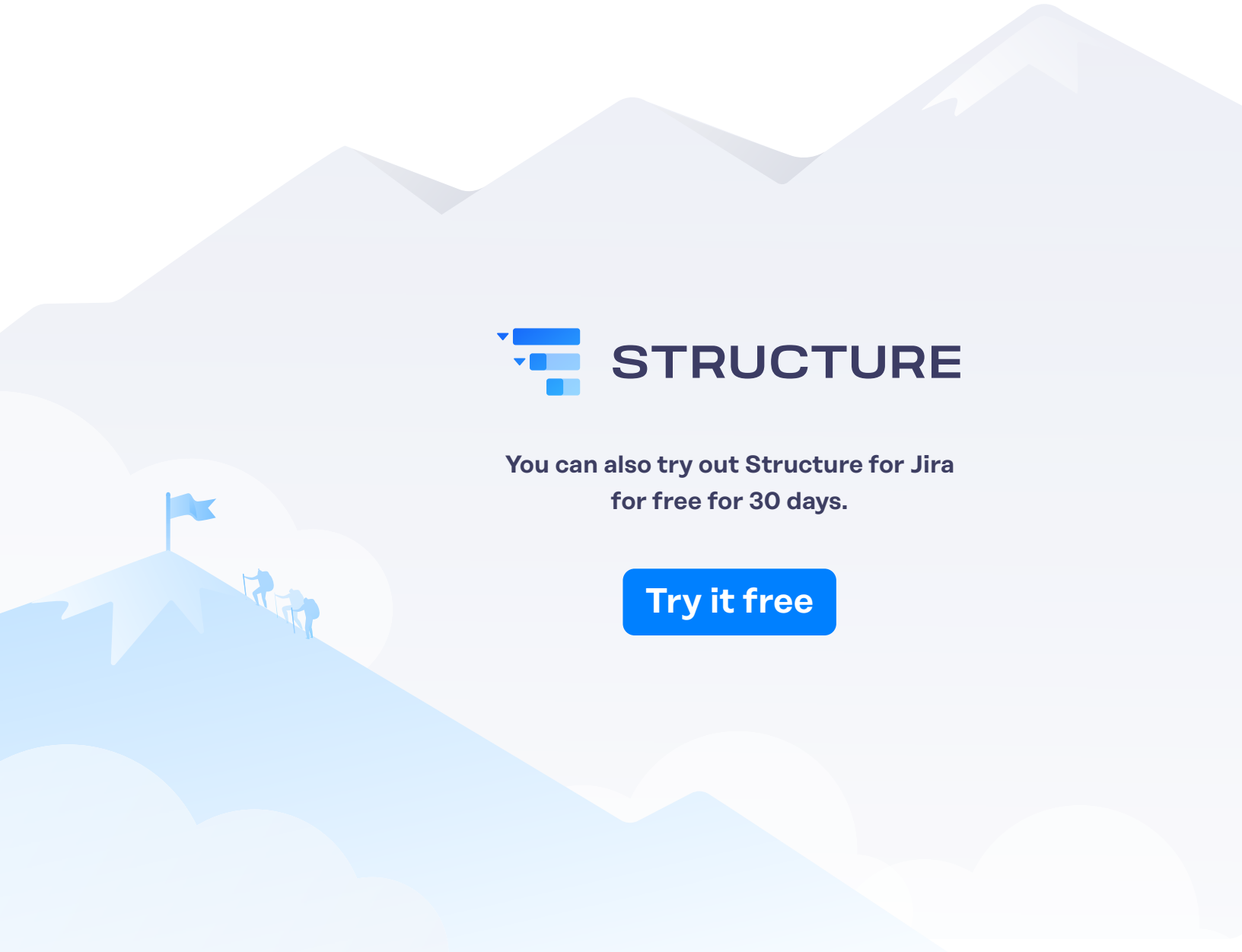
"Depending on where we see problems, we might add a step here or there — a code review step, or a code deployment phase after we started doing software as a service."

They need tools that are flexible enough to allow for adaptations in the process.

Getting to the next level

The process and tools BlueCat had used as a younger, smaller company had to change. It was awkward to sync up spreadsheets and Jira, and made the work difficult to track.

But instead of looking for a revolution that forced bigger workflow upheaval on the company, Arrey looked for ways to support BlueCat's process — as it existed in that moment, as well as the ways in which it was likely to evolve. Using Structure for Jira, they laid out the views that they wanted, automated actions to track tasks, and gave themselves space to change their process as needed.



STRUCTURE

You can also try out Structure for Jira for free for 30 days.

Try it free

Jira apps used in this ebook:

STRUCTURE

GANTT
Structure extension

PAGES
Structure extension

Contact Us

If you want to discuss your own use-case, contact the Structure team!

structure@almworks.com

Structure for Jira is part of Tempo Software, Inc

ATLASSIAN Platinum Top Vendor

Software Reviews GOLD MEDAL 2021 Application Lifecycle Management

Inc. 5000

SCALED AGILE PARTNER PLATFORM