

how to use
**Jira for waterfall
and hybrid projects**

the
perfect blend



old street
solutions

Contents

Introduction	03
<hr/>	
Agile vs waterfall vs hybrid	05
Waterfall	05
Agile	06
Who's using waterfall and who's using agile	07
The hybrid approach	09
<hr/>	
Waterfall and hybrid projects in Jira	10
Jira Software and Jira Work Management	11
Advanced Roadmaps for Jira	14
BigGantt	18
<hr/>	
Conclusion	22

Introduction

Even though Jira is officially an agile project management system, purely agile projects are rare in real life. Most project managers still practice waterfall project management to an extent. And although most development teams have been using agile methodologies to build software for years, business teams such as HR, marketing, and legal haven't.



So, can Jira still work for teams and managers doing waterfall?

The answer is yes, it can

However, Jira is best suited to teams and managers who may be doing waterfall now but want to transition to a more agile way of working in the future.

As it happens, that's what most teams and managers these days want, particularly as the global shift to remote working is forcing organizations to become more agile. But while these organizations may understand the benefits of agile, embracing it is easier said than done for teams who are used to the hierarchies and forward momentum that characterize waterfall. That's why a platform that enables them to continue doing waterfall to a certain degree, and introduce agile practices gradually, is the best fit.

The three platforms available within Jira are designed to facilitate agile collaboration across the enterprise.

Jira Software is set up for software teams using Scrum and Kanban and other agile methodologies.

Jira Service Management (JSM) is designed to bring agile to the IT support team by way of DevOps, an offshoot of agile that unites software development (dev) and IT operations (ops).

And **Jira Work Management (JWM)** enables business teams to connect with the dev and support teams and collaborate on projects in a more agile way.

But what's good about JSM and JWM is that they don't shove agile down the throats of the uninitiated. Atlassian knows that within IT service management (ITSM), the IT Infrastructure Library (ITIL) framework of best practices prevails over agile. And whenever IT ops teams are faced with a big implementation project, they'll tend to default to waterfall because it's easier. Same goes for business teams like marketing. Iterative development of a marketing campaign might be the best way of getting the organization's messaging just right, but

◆ Jira Software

⚡ Jira Service Management

▮ Jira Work Management

permanently pinched marketing budgets and strict deadlines often make waterfall the only way to go.

So, Atlassian allows these teams – the ones that just aren't ready for agile quite yet – to follow their own path. Within Jira there are ways of doing waterfall using either built-in functionality or Atlassian Marketplace add-ons, and you can have either fully agile and fully waterfall projects side by side, or mixed, hybrid projects carefully aligned.

In this white paper, we'll explore the options within Jira for managing waterfall and hybrid projects, and evaluate how these approaches might evolve.



Agile vs waterfall vs hybrid

Let's start by explaining the differences between agile and waterfall methodologies, the advantages and disadvantages of each, and which teams are using them.



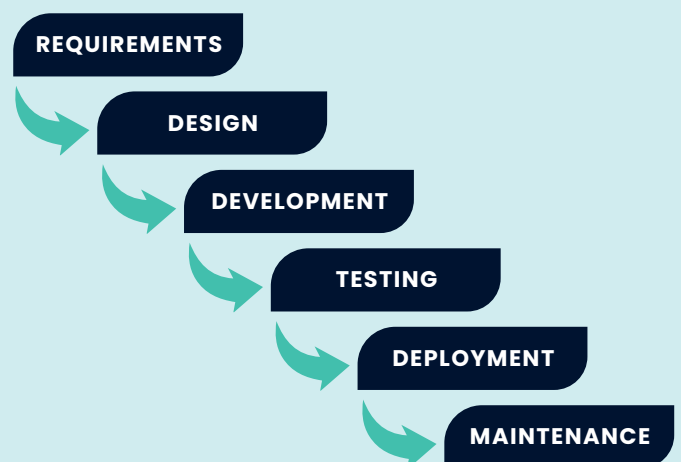
Waterfall

Waterfall is the traditional approach to project management. It is a linear, sequential model of project management whereby projects are divided into pre-defined phases. Each phase must be completed before moving on to the next, because water only falls one way. The typical flow of a waterfall project is:

Requirements > Design > Development > Testing > Deployment > Maintenance.

And these phases are often handled by single-discipline teams operating in silos, rather than cross-functional teams comprising different disciplines working together.

Waterfall is typically used for small projects, projects with unchanging requirements, projects with fixed deployment dates where there is no margin for altering the scope of work, and projects with a lot of dependent tasks. It's generally considered easy to understand and manage, with fewer production issues, and friendly to strict budgets and time frames. The problem is that projects these days are much more complex than they used to be, and waterfall isn't built for it. Sometimes water really needs to fall the other way.

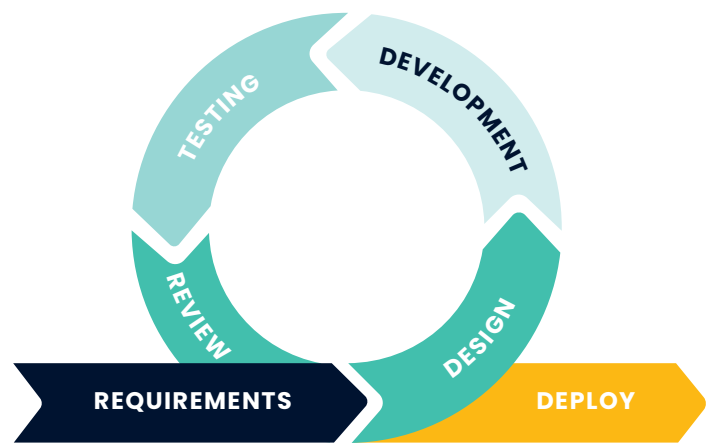


Agile

Agile was born to counter the inflexibility of the waterfall model. In the 1990s, it could take about three years for a working software application to be delivered. But requirements, systems, and even whole businesses can change in that time. It meant that many projects were cancelled partway through or didn't meet the business's current needs once completed. If that happens, it doesn't matter that the waterfall model helped you stick to the budget. All of it is wasted if the product isn't right.

Agile alleviates this risk. Instead of planning everything out in advance and never deviating from it, agile is about development through evolution. Teams are self-organized and cross-functional and collaborate with their customers/end users to get things right. With agile the focus is on getting a minimum viable product (MVP) to the customer as quickly as possible. Importantly, it allows teams to go back a stage and change something, and to adjust projects mid-course when new customer needs arise.

Agile works best for projects where the full set of requirements aren't yet known, as well as projects where the deadlines are adjustable, or in the case of a hard deadline, the scope of work can be changed to meet it. It's also best for teams with direct access to customers, projects without strict budgets, and teams that are trusted to do the jobs they know how to do rather than being micro-managed by their superiors.



Who's using waterfall and who's using agile?

Most small-to-medium-sized software development teams that are free to think quickly, be flexible, and work independently without managerial interference, use agile because it's better for their customers and users. They shunned the waterfall model years ago because for them it just doesn't work.

To be honest, in a lot of contexts, they're right. Research has consistently shown that agile projects are more likely to succeed than waterfall projects. In 2018, the Standish Group Chaos Study found that agile projects are twice as likely to succeed as waterfall projects, as well as a third less likely to fail.

However, while agile is the norm at the team level, go further up the organizational hierarchy and it's a different story. Most project managers still use waterfall to a certain degree, mainly due to contracts, laws, and budgeting. If you have to concern yourself with delivering something by a certain date on an immutable budget, then there isn't a lot of room for flexibility. In addition, many organizations still rely on top-down hierarchical management, with leadership committed to keeping tight control on what's being built. Leaders like this are often resistant to letting teams self-organize.

Agile was also considered a software team thing only, largely because that's who it was invented for. Although the benefits of adopting agile practices are getting recognized by non-software teams, it's still a new concept for many of them. Legal teams, for example, have been doing things in much the same way for the past 100 years. Expertise is siloed by practice area and work is planned out in full before anything happens. Similarly, finance and procurement departments are used to working with guaranteed deliverables for a fixed price. For these teams, traditional waterfall management is baked into their processes.

But then 2020 came along and teams everywhere were suddenly displaced from their offices to work remotely. Now, distributed teams are becoming the norm across most industries and disciplines, which makes agile self-organization over hierarchical waterfall management a much easier fit. The boom in remote working is one of the main reasons why agile adoption is accelerating.



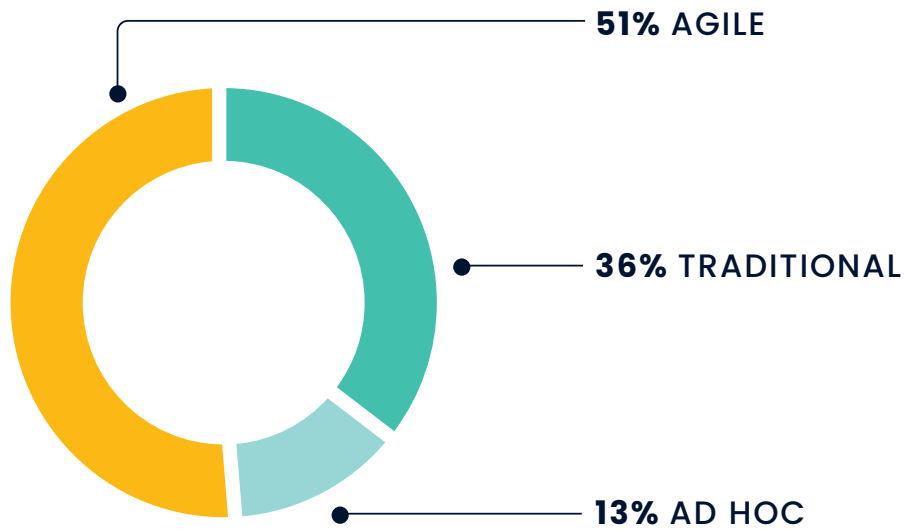
“

*Teams everywhere
were suddenly
displaced from their
offices to work remotely*

”

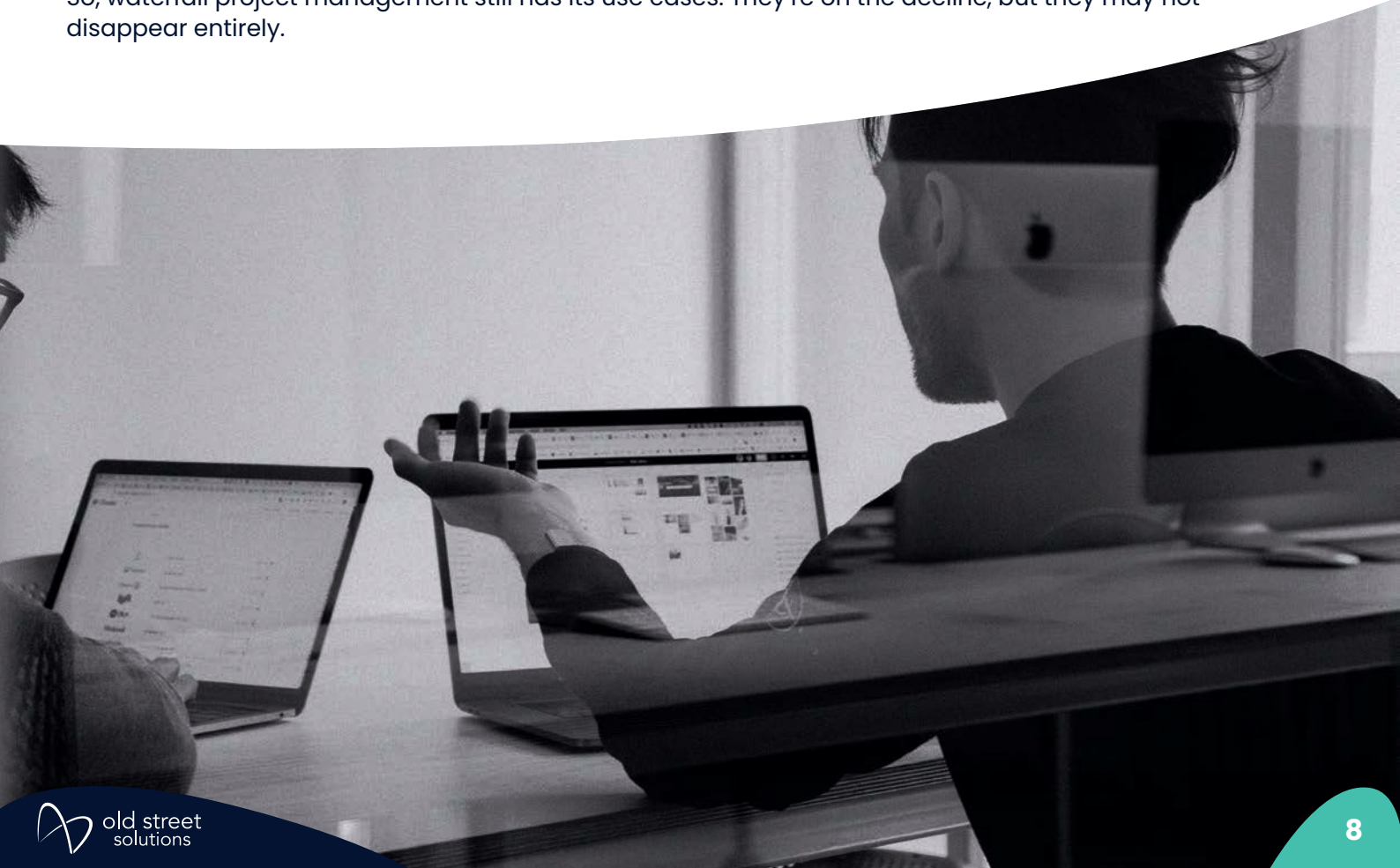
For example

The 4th Annual State of Agile Marketing Report found that agile adoption among marketers had jumped from 42% in 2019 to 51% in 2020. They noted that it was the first time that over half of marketers had labelled themselves as agile.



But even though agile is the way to go in most situations, it's not necessarily in every situation. If a task doesn't usually see a lot of waste – perhaps because work doesn't need to be tested or redone repeatedly, customer feedback isn't an important component, or changes to the deliverable are very easy to make – then the iterative aspects of agile might not add much value.

So, waterfall project management still has its use cases. They're on the decline, but they may not disappear entirely.



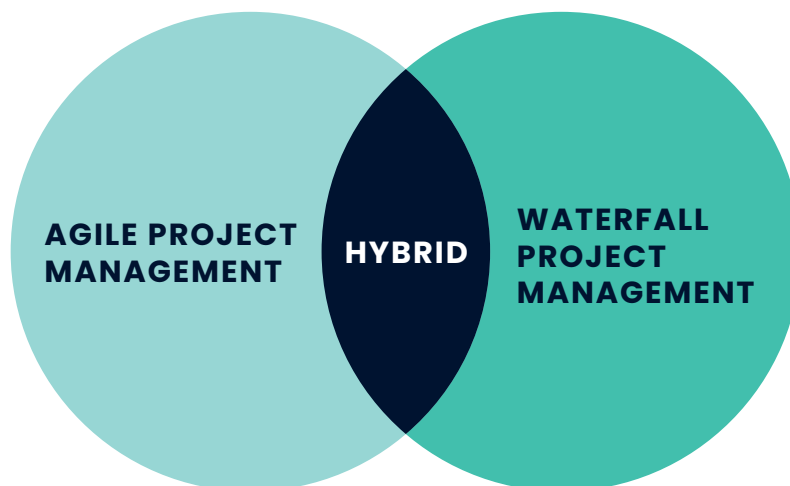
The hybrid approach

There's then the option of trying to get the best of both worlds. We say "trying" because it can be problematic and doesn't always work. Waterfall and agile are two very different approaches, so combining them has to be done carefully. Often we see organizations attempt a hybrid approach but it just turns into waterfall.

However, combining waterfall and agile is possible and does have its upsides. It's arguable that customers prefer a bit of both: they want working software as quickly as possible (agile) but they'd also like to define the budget and time frame well in advance (waterfall). If your customer's insistent on both, perhaps a hybrid model is the way to go.

Erick Bergmann and Andy Hamilton said that the agile-waterfall hybrid model would typically involve the following:

- **Software development teams using agile**
- **Project managers and hardware development teams using waterfall**



An alternative way of combining them would be to plan with waterfall and execute with agile. So use waterfall to establish requirements, plan, and design, and use agile to develop, test, and deploy. You might also consider using waterfall at both the project and enterprise level, but agile for individual teams.

What we would say is that for a hybrid approach to work long-term, there needs to be a clear line between who's doing agile and who's doing waterfall.

What we often see is organizations cherry-picking certain agile components and incorporating them into their usual waterfall approach. That isn't a successful hybrid methodology. That's an organization in transition, moving from waterfall to agile gradually. If you view it as the latter, it works, as long as it's recognized that an organization can't be in transition forever and will eventually need to make the leap to agile. But if it's viewed as hybrid and the intention is to carry on that way indefinitely, then you're really just waterfall pretending to be a bit agile. For example, it's easy to think you're agile if you're doing sprints, but Scrum is a very specific and well-defined agile framework, and the moment you start dropping bits out of it, you start to lose all the benefits of it.

So, it's important that you dig into why you're taking a hybrid approach. Were you previously doing waterfall but the changing times have forced you to incorporate agile into your working? If so, look at hybrid as a stepping stone to full agile. And that's where Jira can really help.

Waterfall and hybrid projects in Jira

As mentioned at the start, Jira is an agile project management platform. However, Atlassian wants it to be as versatile as possible and understands that many organizations and teams aren't ready for full agile. When it launched Jira Work Management earlier this year, Atlassian stated that it had succeeded in making Jira suitable for all teams, whatever their style of working. That includes business teams such as HR and marketing, who may take a more traditional waterfall approach to their work, but want to become more agile in order to collaborate better with the software and support teams who are agile already.

To that end, Atlassian has been adding more and more features to accommodate waterfall and hybrid teams over the past year. Now you can even use standard Jira to manage a waterfall or hybrid project if you wanted to, although the options are limited. The best way of doing waterfall and hybrid in Jira is by using Jira's core functionality in conjunction with an additional marketplace app.

So in this next section, we're going to talk about three ways of doing waterfall and hybrid projects in Jira:

- 1. Using Jira's core functionality such as epics, boards, and releases**
- 2. Using Advanced Roadmaps, a Jira Software Cloud Premium and Data Center feature**
- 3. Using BigGantt by SoftwarePlant, an Atlassian Marketplace add-on**

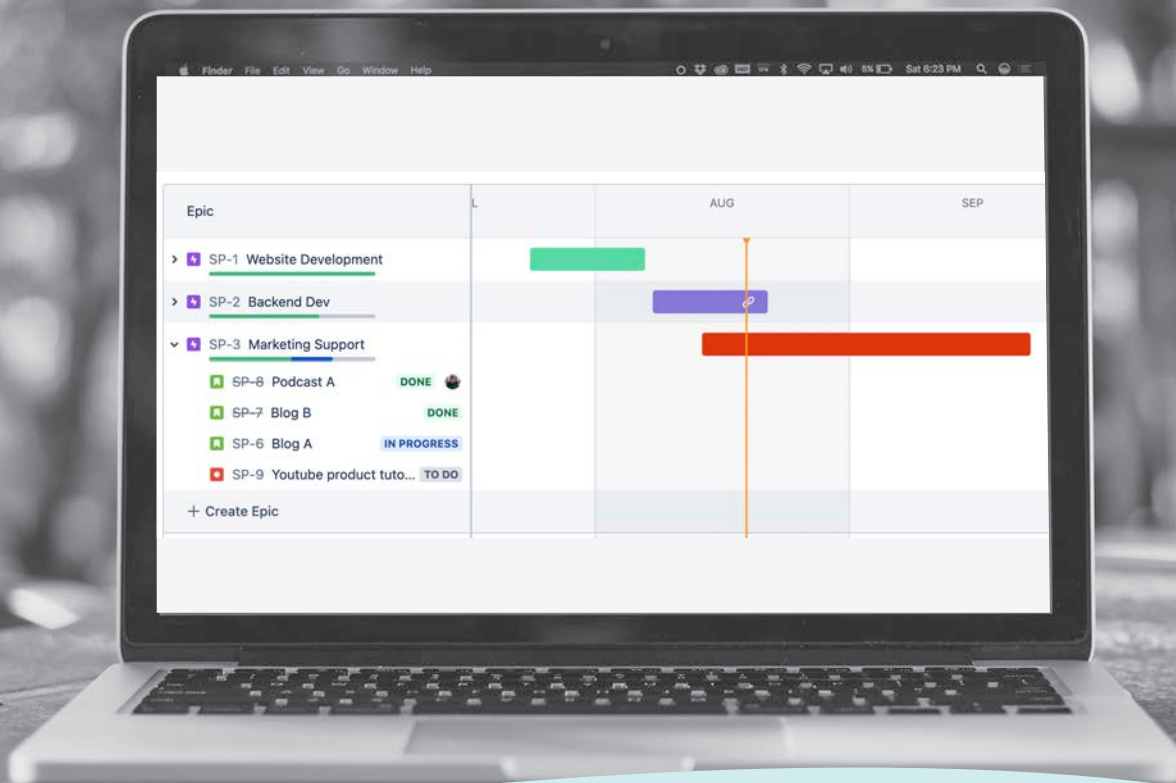


Jira Software and Jira Work Management

The add-ons we'll review are powerful, but without intentionally designed projects in native Jira alongside them, it will be difficult for waterfall and hybrid teams to leverage the extra functionality.

Jira Software and Jira Work Management come with some great out-of-the-box tools that your waterfall and hybrid teams will want to take advantage of first, before looking at add-ons to fill any gaps. So let's start with these.

Hierarchy



When managing a project, the ability to see a work breakdown structure (WBS) and understand a hierarchy of work to be completed is the first thing that matters. Jira ships with a three-level issue structure that serves as the foundation for your projects. This structure is: **epic > story-level > sub-task**.

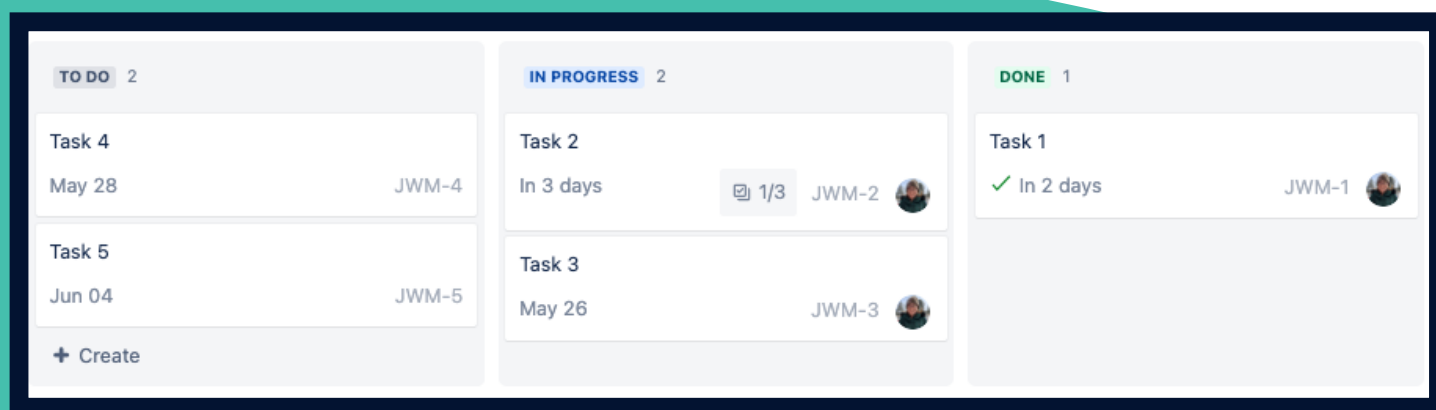
Most of the work done in Jira is at the story level. The story level is where tasks, bugs, stories, and most custom issue types live. These are broken down into specific issue types to allow them to have separate workflows and processes. If you're a hybrid team, you may want to make use of concepts like user stories and spikes combined with more traditionally waterfall issue types like tasks, deliverables, and risks. Much of the time these story-level issues are not connected to any other issues and can stand alone.

Story-level issues can optionally be broken down into sub-tasks. Sub-tasks are children of your story-level issue types, and cannot be created without a parent issue. These can be used to break out tasks into smaller pieces of work or assign parts of the work to different people.

Above the story level are epics. Although epics are an agile concept, they are useful to waterfall and hybrid teams because they can be used in many different ways, and really just represent larger buckets of work. A single epic can have many tasks linked to it, which makes them great for grouping issues together in a way that is easy to keep track of. With epics, you get some extra board reporting as well, and the ability to track basic progress across issues.

This three-level structure alone allows for flexibility in how teams in Jira manage their work, and depending on the complexity of your project management process, this native Jira hierarchy may be sufficient.

Boards



Both Jira Software and Jira Work Management ship with the ability to view your work on a board. In Jira Software you have the option to choose between Scrum and Kanban boards, and in Jira Work Management you'll be using a Kanban board.

Generally, non-agile teams should avoid Scrum boards because they are designed to be used by teams running sprints, and many of the structures that make them good for agile teams make them difficult to adapt to if you're not following those processes.

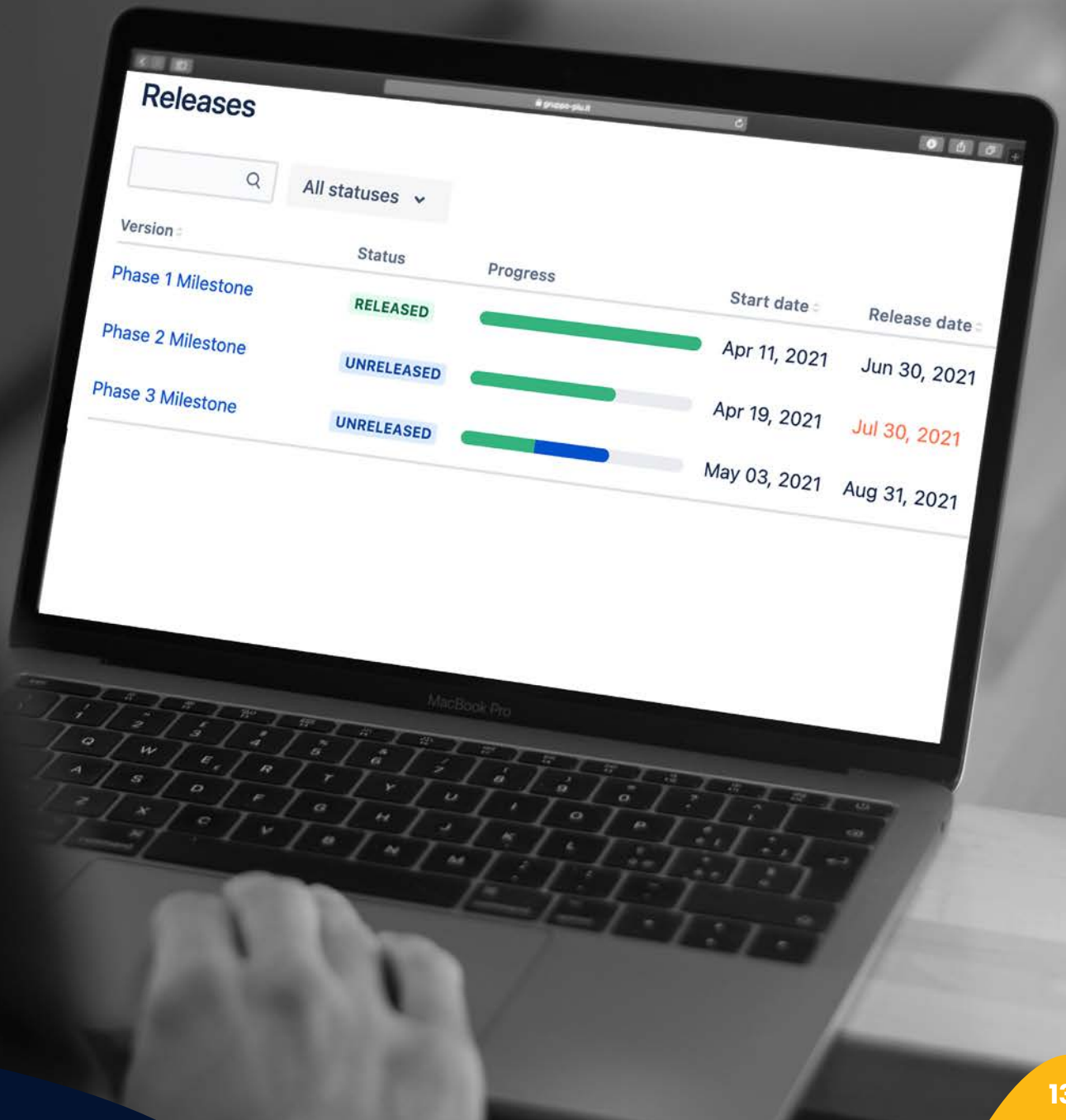
On the other hand, as a hybrid or waterfall team, Kanban boards will be your best friend. Although Kanban boards are an agile tool, unlike Scrum boards, they are extremely flexible. Offering a versatile visual representation of your work, they have columns representing one or more workflow states, where you can drag and drop your issues to show how the work is progressing.

A single Jira project can have more than one associated board, allowing you to visualize the information in different ways. For example, the project teams executing the work at the task and sub-task level could have their own boards, while the project managers could have separate boards to oversee the larger buckets of work (i.e. epics). As you introduce additional project management tools into Jira, it is likely that your teams will continue to use boards to manage their work, while your project managers move into the customized app views that we'll discuss below.

Releases

Finally, releases (aka versions) are an often overlooked but extremely valuable tool in Jira Software. Releases are represented in Jira using the fixVersion and affectsVersion fields. While designed for development teams to track software releases, the functionality is fairly generic, which means you can use a version to represent project phases, major deliverables, or key dates to drive work. For your teams working in a traditional project management structure, the ability to represent phase gates or major dates in Jira itself is going to be incredibly useful.

In native Jira, releases are created and managed on a per-Jira-project basis. Each issue can have one or more versions, and a version can have a name, start date, end date, and description.

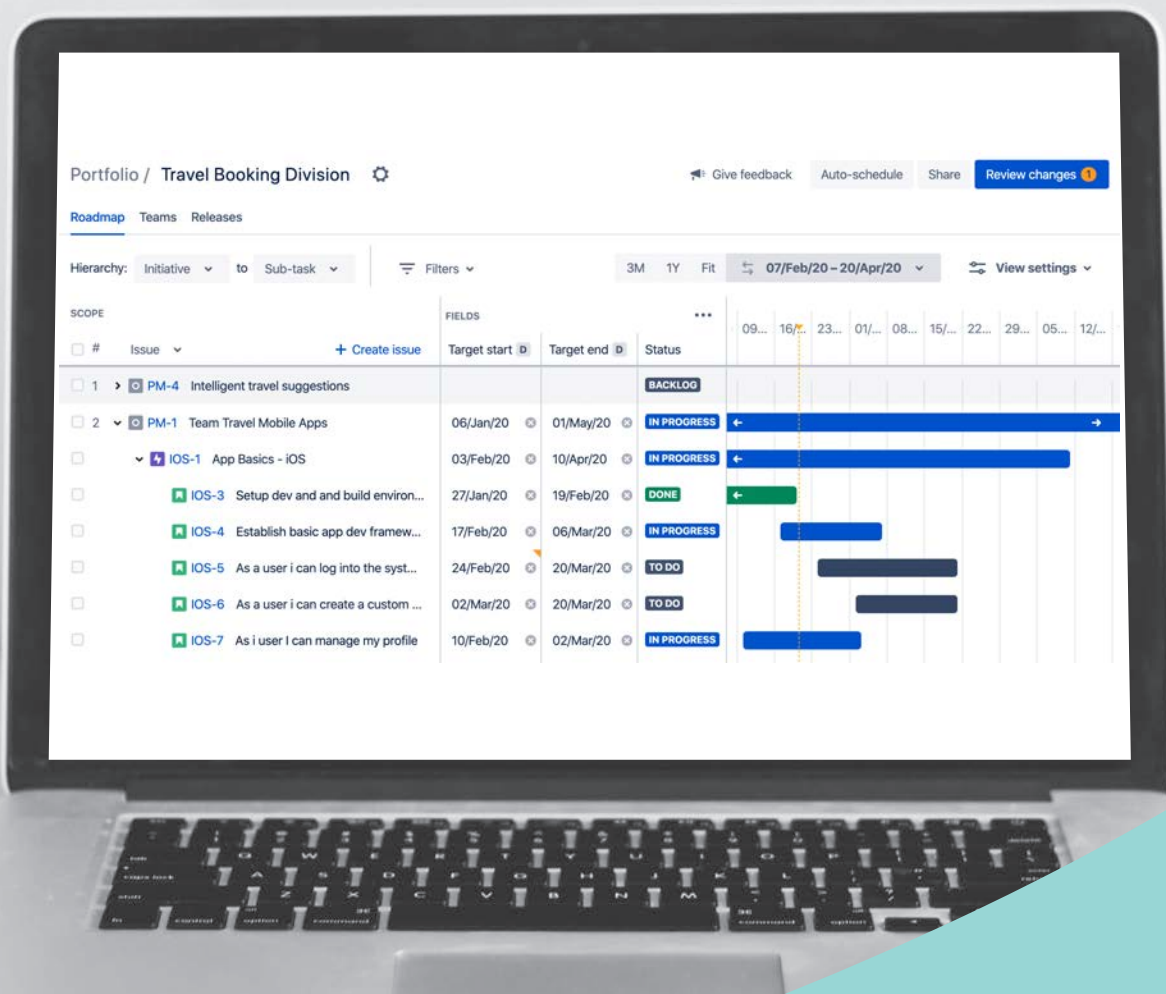


Advanced Roadmaps for Jira

Advanced Roadmaps for Jira (originally known as Portfolio for Jira) builds on the native Jira structures described previously. Currently, Advanced Roadmaps ships with your Jira Software Cloud Premium or Data Center subscriptions. If you are on a Standard plan for Jira Cloud or using a Server instance, this app is not available and you may want to look to BigGantt (which we'll talk about later).

Advanced Roadmaps provides improved views of your issue data and the option to add hierarchical levels above an epic, and enables you to auto-schedule your work based on scope, time, and team capacity. Although previous versions of Advanced Roadmaps were strictly for agile teams, just like Jira it has come a long way to be process-agnostic.

Now, its improved flexibility means that if you are running a hybrid model or even strict waterfall, you can still use Advanced Roadmaps to plan your projects and programs. Some of its features allow extra visibility that can help improve agility in your project management process, which makes it great for hybrid teams transitioning to a more agile way of working.



Hierarchy

Advanced Roadmaps provides the ability to add extra hierarchical levels above an epic, such as an initiative, an even larger bucket of work or an overarching work/business goal. You can actually add as many hierarchical levels above an epic as you want, and while initiatives are a common example, this can be modified to respect whichever work breakdown model you are using.

These extra levels are created by assigning specific issue types to each level above an epic. For example, initiatives live in Jira as their own issue type, and can have their own process for project requests, approvals, or other data points you might want to store on them.

If you are also looking to improve how projects are managed by a project management office (PMO) or product organization, representing these hierarchical levels as issue types is going to be one of the biggest advantages of using Advanced Roadmaps over alternatives on the marketplace.

Once you've got your issue types and hierarchy identified, you can set up your plan.

Plan definition

Advanced Roadmaps is broken down into plans. A plan can represent any body of work that you want to track – most often a singular plan represents a program or even a large project.

Plans are defined simply by pulling in Jira issues using boards, Jira projects, or Jira Query Language (JQL). Because of this flexibility in how plans are defined, waterfall teams can pull in issue data for only the relevant work, rather than everything in a single Jira project.

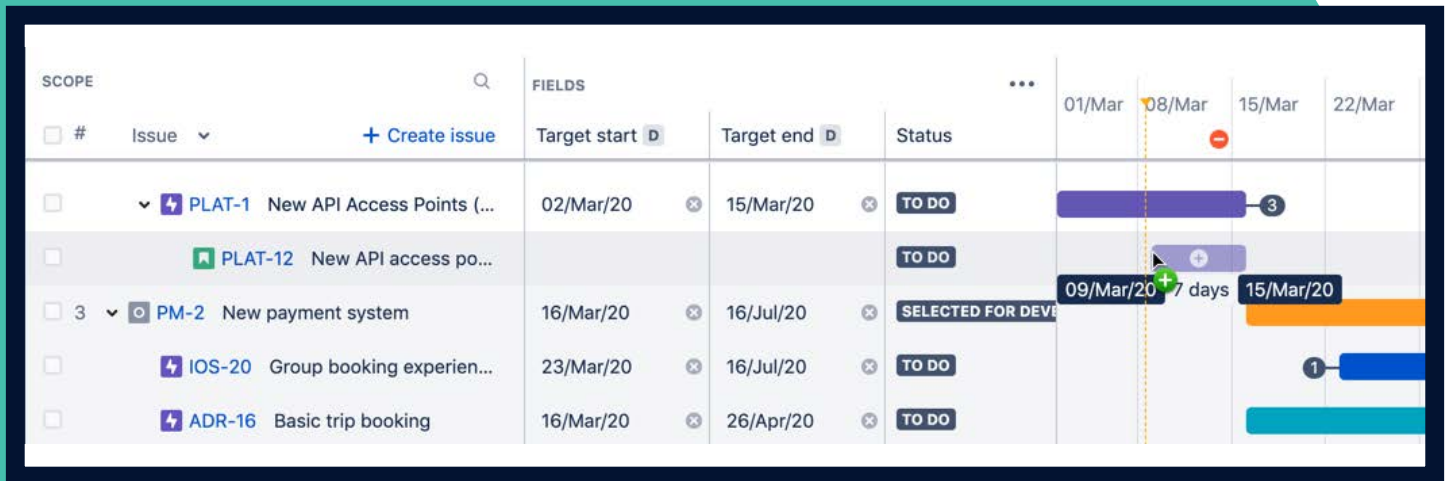
A single Advanced Roadmaps plan displays these issues in a Gantt view, with an easily navigable work breakdown structure, and additional reporting on dependencies within those issues. In each plan, you can also create and manage teams.



“
Once you've got your issue types and hierarchy identified, you can set up your plan

”

The scope view



The bread and butter of Advanced Roadmaps is the scope view. This allows project managers to view issues included in your plan, and also lets you display custom fields, roll up the estimated work to higher levels, and expand the hierarchy to see your issues in their full context.

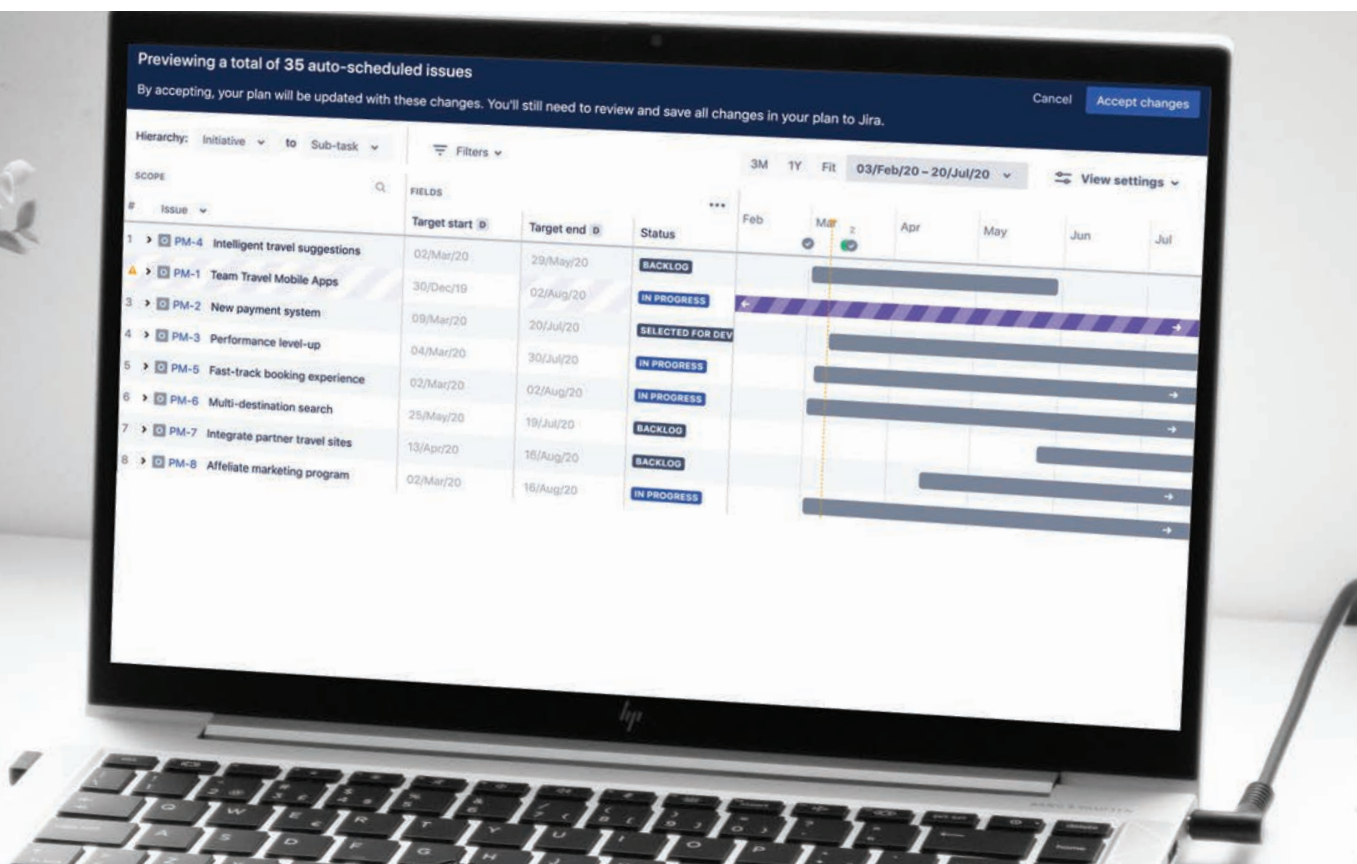
The scope view is highly customizable and is where most of your time in the tool will be spent. The flexibility here means that it will be useful to teams using any flavor of waterfall or hybrid project management. The view is broken into two sections: the top timeline view that displays a Gantt chart, and the bottom issue scope list that provides your collapsible work breakdown structure.

The Gantt view works like most, showing bars based on issue start and end dates. If you're using the auto-scheduler described in the next section, you can even see timeline slippage visually represented. This view allows for drag and drop of issue start and end dates and can be filtered, grouped, and even color-coded.

The bottom half of the scope view is where your custom fields are listed based on the rank of the issues. You can customize which hierarchical level you are looking at, and expand lower levels to see a full breakdown. This then becomes one of the most important views for your project managers in terms of understanding the status of a project.



The auto-scheduler



Arguably the most powerful feature in Advanced Roadmaps, the auto-scheduler gives you visibility into how long work will take to be completed based on constraints you've already entered into Jira. This tool applies Jira concepts to the project management iron triangle, allowing you to control as many of the aspects as you want to determine target dates for work completion at any level. It also allows you to dynamically schedule your issues – set one or more of the variables of release/team/issues to be flexible, and the auto-scheduler will offer a plan for your work.

By way of example, if you have a specific set of deliverables that need to be in the next milestone, but you don't care which team works on it or when the work is completed, you can set the milestone release dates to be flexible, and then auto-schedule to see which teams have the capacity to complete the work next.

This feature is extremely powerful if you have some amount of flexibility in your planning. Combine the auto-scheduler with the ability to create isolated scenarios in Advanced Roadmaps, and PMOs can try out different options without impacting your official timeline. Although this may be a little too uncertain for some waterfall teams, being able to see what-if scenarios, even if you don't commit them to Jira, can help you transition to a more agile project planning model if you're a hybrid team.

“

This feature is extremely powerful if you have some amount of flexibility in your planning

”

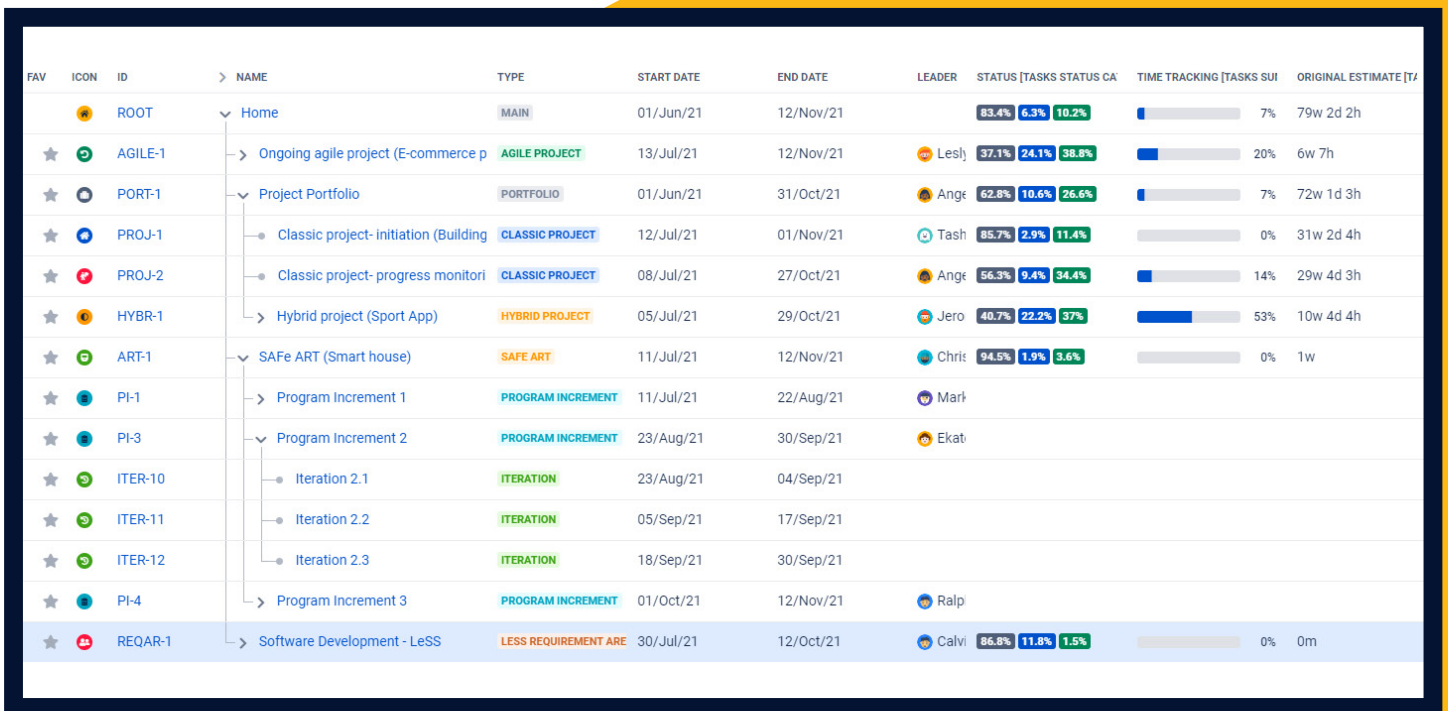
BigGantt

A recommended project management Jira add-on from the Atlassian Marketplace is BigGantt. Introduced by SoftwarePlant, BigGantt is another strong app that can be used to help manage waterfall and hybrid projects in Jira.

SoftwarePlant offers three versions of the tool – BigGantt, BigPicture, and BigPicture Enterprise. We'll focus on BigGantt because its simplicity makes it well suited to waterfall or hybrid teams. BigPicture provides some additional modules, and BigPictureEnterprise even more, tailored to scaled agile organizations.

BigGantt provides the Gantt views you might expect from the name, with some basic but powerful resource and risk management views.

Box definition



FAV	ICON	ID	NAME	TYPE	START DATE	END DATE	LEADER	STATUS	TASKS	STATUS CA	TIME TRACKING	TASKS SUI	ORIGINAL ESTIMATE	TR
		ROOT	Home	MAIN	01/Jun/21	12/Nov/21		83.4%	6.3%	10.2%		7%	79w 2d 2h	
★		AGILE-1	Ongoing agile project (E-commerce p	AGILE PROJECT	13/Jul/21	12/Nov/21	Lesl	37.1%	24.1%	38.8%		20%	6w 7h	
★		PORT-1	Project Portfolio	PORTFOLIO	01/Jun/21	31/Oct/21	Angt	62.8%	10.6%	26.6%		7%	72w 1d 3h	
★		PROJ-1	Classic project- initiation (Building	CLASSIC PROJECT	12/Jul/21	01/Nov/21	Tash	85.7%	2.9%	11.4%		0%	31w 2d 4h	
★		PROJ-2	Classic project- progress monitori	CLASSIC PROJECT	08/Jul/21	27/Oct/21	Angt	56.3%	9.4%	34.4%		14%	29w 4d 3h	
★		HYBR-1	Hybrid project (Sport App)	HYBRID PROJECT	05/Jul/21	29/Oct/21	Jero	40.7%	22.2%	37%		53%	10w 4d 4h	
★		ART-1	SAFe ART (Smart house)	SAFE ART	11/Jul/21	12/Nov/21	Chri	94.5%	1.9%	3.6%		0%	1w	
★		PI-1	Program Increment 1	PROGRAM INCREMENT	11/Jul/21	22/Aug/21	Mar							
★		PI-3	Program Increment 2	PROGRAM INCREMENT	23/Aug/21	30/Sep/21	Ekat							
★		ITER-10	Iteration 2.1	ITERATION	23/Aug/21	04/Sep/21								
★		ITER-11	Iteration 2.2	ITERATION	05/Sep/21	17/Sep/21								
★		ITER-12	Iteration 2.3	ITERATION	18/Sep/21	30/Sep/21								
★		PI-4	Program Increment 3	PROGRAM INCREMENT	01/Oct/21	12/Nov/21	Ralp							
★		REQAR-1	Software Development - LeSS	LESS REQUIREMENT ARE	30/Jul/21	12/Oct/21	Calvi	86.8%	11.8%	1.5%		0%	0m	

Just as Advanced Roadmaps is broken down into plans, BigGantt is broken down into boxes. These can represent any large collection of projects or programs that you're looking to track.

In contrast to Advanced Roadmaps, boxes are intended to have distinct start and end dates and display as blocks such as project phases or quarters. The ability to have dates here is an attractive feature to waterfall and hybrid teams, where projects are often more strictly date-driven, or broken down into very defined phases.

Having a box with hard dates per project or phase to represent your phase gates can be particularly useful.

The source of a box is similarly tied to projects, saved filters, board, or JQL written directly in the box configuration. For teams who have a combination of Jira and Trello data, boxes can also connect to Trello boards.

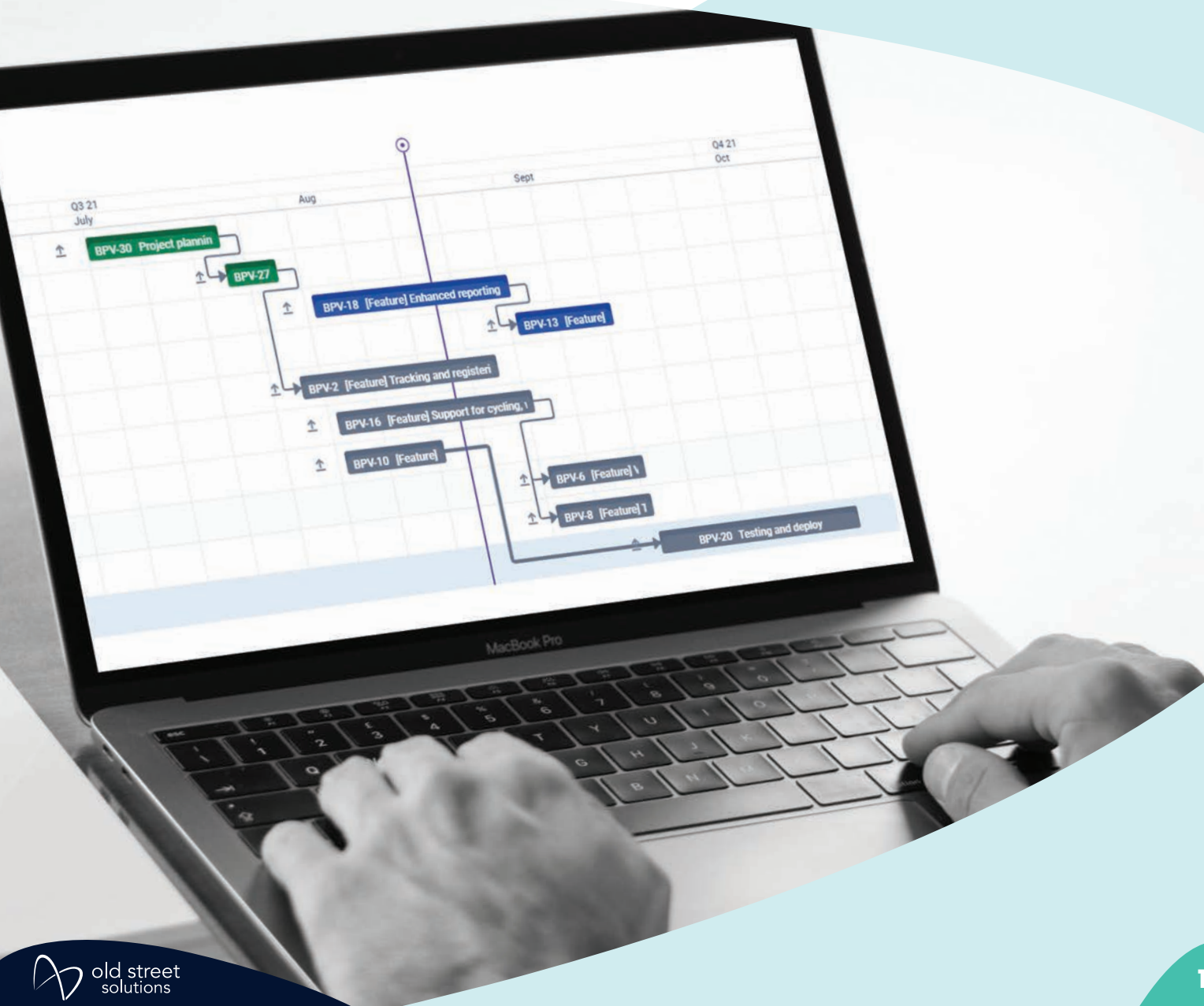
Gantt view

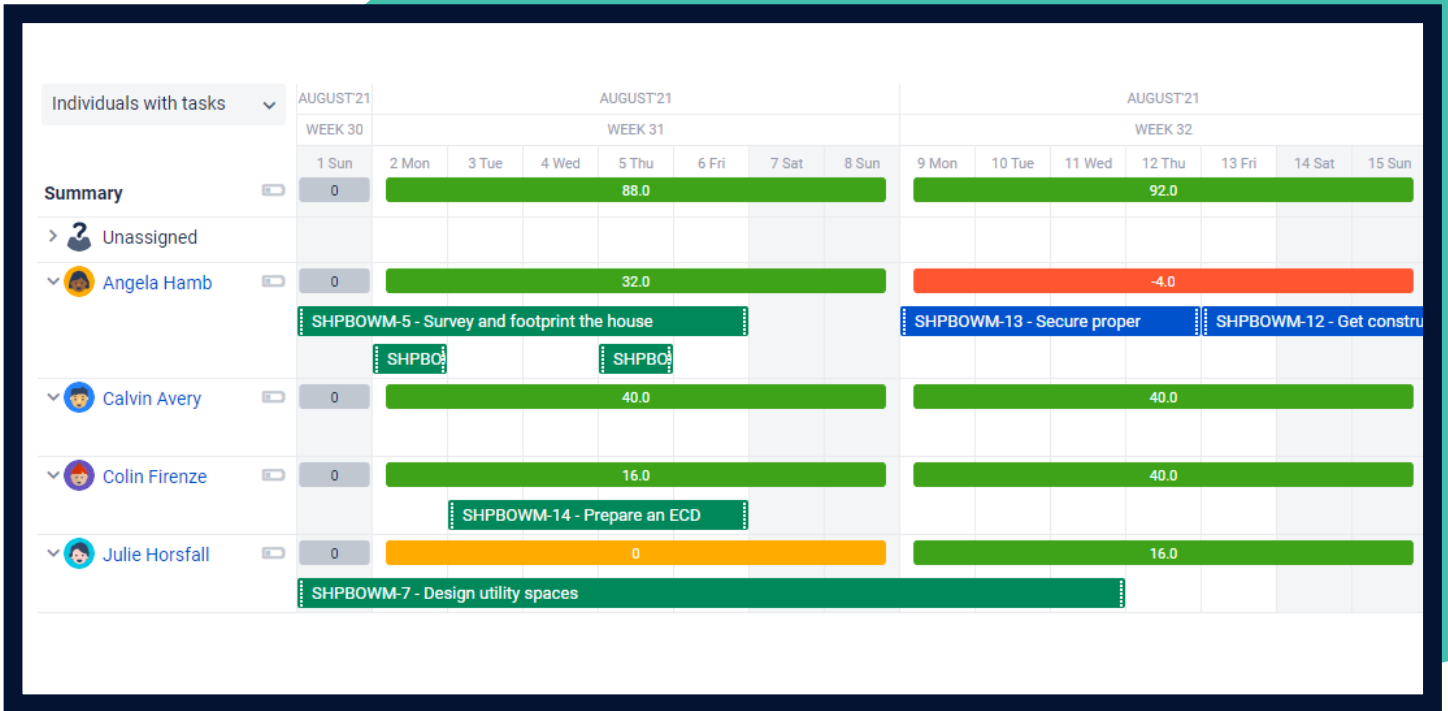
Once a box is defined, the foundational feature of BigGantt is the Gantt view. The left side of your screen will display the work breakdown structure, with the associated Gantt chart on the right. The benefit of this split, as opposed to the top/bottom division of Advanced Roadmaps, is that it can be easier to see how a specific piece of work is driving the roadmap.

The hierarchy in BigGantt is fairly informal. You can drag and drop to create a hierarchy within the tool and organize your issues ad-hoc in whatever way makes the most sense in the specific situation. If you don't have a standardized process across projects, this flexibility will be a lifesaver as it allows each project manager to manage their projects how they see fit.

This Gantt view also shows dependencies based on issue links, as well as baseline dates. Baselines can be set, then a very straightforward original vs. actual timeline can be reported on, and milestones added to clearly see major dates in your timeline.

This work breakdown structure can also be viewed in issues directly. When enabled, the hierarchy that an issue belongs to will display in the issue detail view, allowing users to understand the context of the work they're doing without leaving the issues. This feature is very powerful and allows users who are not intimately familiar with BigGantt to still benefit from its usage.





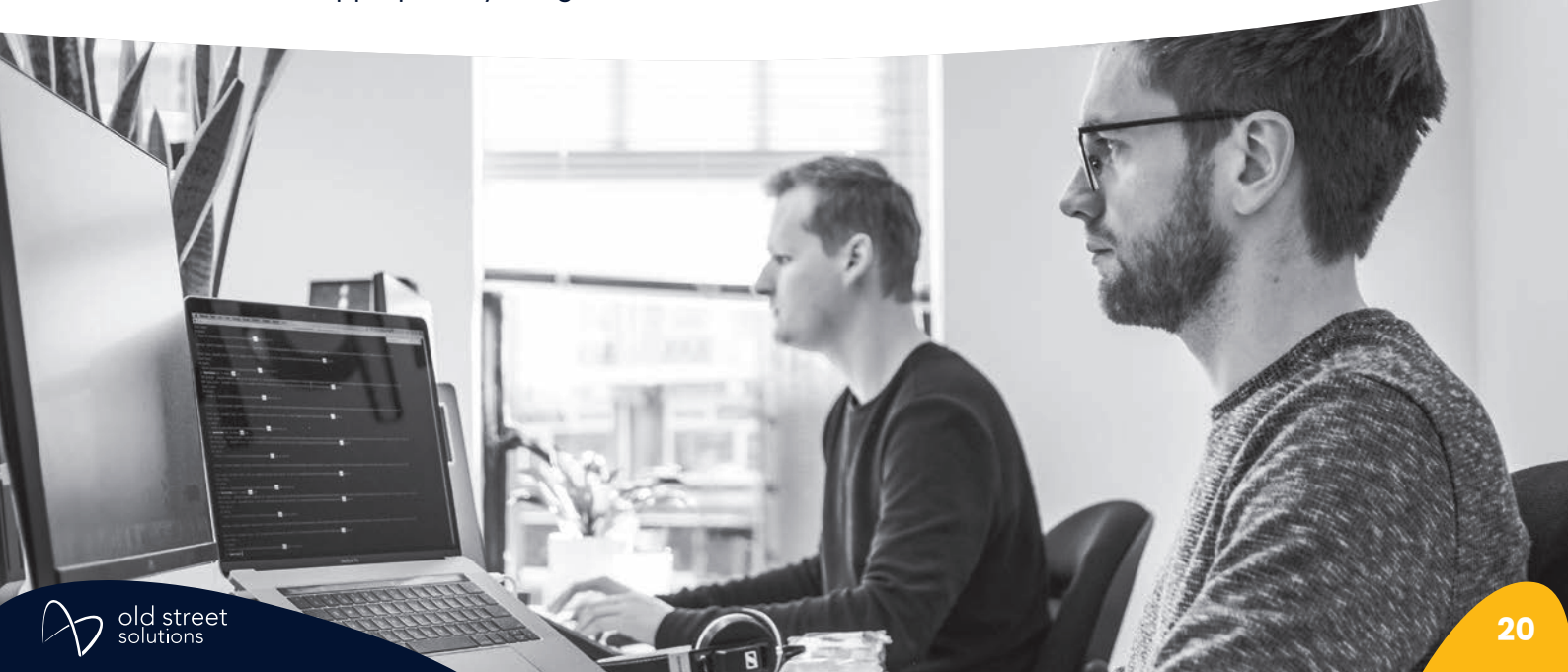
Resources

Resource management is an extremely common request for PMOs. The ability to manage team members across projects is important to ensure that your team members aren't overloaded. Because contributors to waterfall projects are more likely to be in project-based teams that change regularly rather than longstanding teams, gaining an understanding of when and where people are committed to work is essential.

So, create teams and assign them to issues in Jira or BigGantt and visualize capacity based on individual assignees. In the resources module, your estimates and assignees are used to determine if your team members are under-allocated, over-allocated, or appropriately assigned work.

This view is great if you want to find out which resources will be available at a certain time. Using the resources tab as a single source of truth can allow project managers to more accurately gauge their resource pool.

Planning by team is possible, too, as teams can be created and then have individuals added as members with specific start and end dates. Simply provide capacity for each team member, and the team capacity will be the sum of those numbers. If your teams are project-based and commonly change, then this feature will be especially useful.



Risks

Another commonly requested traditional project management feature we haven't discussed yet is risk management. SoftwarePlant's risk module provides the visualization that waterfall teams are used to. In Jira, it is possible to create an issue type to represent risks and assign a workflow to represent whichever risk management process you use. However, you are limited in how you can visualize those risks, just as you are with any other issue type if you're using Jira alone. BigGantt offers a more robust view, letting you see your risks in a heat map or table.

The default risk register comes with two data points to determine risk level: risk probability and risk consequence. Although BigGantt ships with the standard low/medium/high, you can customize these values. For example, teams who use ROAM to manage their risks can use Resolved/Owned/Accepted/Mitigated as their values.



“

BigGantt offers a more robust view, letting you see your risks in a heat map or table

”

Conclusion

Even though Jira was designed to be an agile tool, it comes with a lot of functionality for managing projects that is entirely suitable for teams using other methodologies. Indeed, don't be discouraged by the agile language often used in Jira and the other apps we've talked about in this white paper. Many of the tools and features can be used just as successfully by waterfall and hybrid teams even if they sound like they're just for agile ones. Epics and releases can serve as the foundation for more complex planning processes, while Kanban boards are great for showing work in progress however that work is being managed.

Ultimately, every team is different, even the agile ones, and how you manage your work in Jira will depend on what your priorities are...

- **If you don't need much in the way of higher-level views or long-term planning, you can probably just stick to standard Jira, without needing to branch into add-ons.**
- **If you're looking for a tool that will seamlessly transition into agile practices as your team does, then Advanced Roadmaps can facilitate that growth.**
- **If you're most comfortable with traditional waterfall views and looking to manage resources and risks, BigGantt will provide what you need.**

What we would say is that Jira is best suited to teams that want to become more agile over time, or want to adopt a hybrid model with certain teams doing waterfall and others doing agile, as long as the line between them is clear. The overarching benefit is that all your teams end up working in the same platform, making it easier to collaborate and communicate, whatever your style of working. And getting everyone in an organization on the same page is essentially what Atlassian is trying to achieve by making Jira suitable not just for software teams, or agile teams, but for all.

Old Street Solutions

is an Atlassian Platinum Solution Partner that makes software to simplify Atlassian's powerful platforms.

Old Street aims to empower all Jira and Confluence users, from programmers to business users to beginners, to create and collaborate more easily, and manage projects with more competence. Old Street's apps are about eliminating barriers and bureaucracy so that everyone can be more useful to their teams.

In 2020, its flagship app, Custom Charts for Jira, was the fastest growing paid app in the Atlassian Marketplace.



old street
solutions



Platinum
Marketplace Partner



Custom
Charts



External
Share



Agile Planning
Boards

