

THE ENGAGED ENTERPRISE'S GUIDE TO SCALING AGILE WITH JIRA ALIGN

PART 1 : SCALING FROM THE BOTTOM UP



INTRODUCTION

“Going Agile” is old news. The concepts are well established in software development and becoming more so in many other industries. And, it’s proven itself to be a highly effective means of streamlining project workflows in many different circumstances. But it’s no longer the ground-breaking revolution it once was.

“Scaling Agile”, on the other hand, is a relatively new concept that many organizations are considering, and some have attempted. They quickly find, however, it’s not as simple as forcing more teams to have a standup every morning or use a Kanban board. There are fundamental changes required to process, culture, and even organizational definitions of success, all of which present challenges at different stages of the Agile journey.

In this two-part series of whitepapers, we’re going to dive deep into how organizations can scale Agile most effectively. It requires both a bottom-up and a top-down approach, the former starting at the individual team level and the latter starting with the C-suite.

As we’ll see, scaling Agile can be likened to a building construction project. It has to start with a strong foundation, there’s a particular set of skills required — including how to choose and use the right tools — and it pays to follow a quality set of plans. Of course, every step of the process requires hard work, too.

These whitepapers are for those rare few organizations that are willing and able to put in the hard, sometimes painful, but ultimately rewarding work required to effectively implement Agile at enterprise scale.

Here's what we'll be covering in Part 1, Scaling From the Bottom Up:

Section One: The Foundation You're Building On

- What does it really mean to scale Agile?
- Why should you care about scaling Agile?
- Do you have to adhere to a particular framework for scaling Agile?

Section Two: Key Concepts to Guide Your Build

- Teams vs Teams of Teams
- Agile Roles at Scale

Section Three: A Blueprint for Success

- Success Pattern: Coordinated Planning and Execution
- Success Pattern: Visibility into Everything for Everyone
- Success Pattern: Continuous Improvement in Planning and Forecasting
- Success Pattern: Refocus on Business Value from Every Angle

SECTION ONE:

THE FOUNDATION YOU'RE BUILDING ON

To lay a strong foundation, we'll start by covering some basics you may already (think you) know. Bear with us, though, because even one missing brick can eventually bring your whole building project down.

What does it really mean to scale Agile?

Agile methodologies (Scrum, Kanban, XP, etc.) focus on individual team planning and delivery activities. Each methodology stipulates roles, ceremonies, and reports to ensure valuable products are incrementally and continually being delivered. For the individual team, these approaches have a strong history of success because they offer impressive benefits, including reduced time to market, nimbleness in the development process, and greater transparency internally.

Still, Agile practitioners have historically struggled to scale the practice across multiple teams and plan work at a higher level. As Agile teams mature and grow, they often have difficulty with:

- Tracking large initiatives that combine multiple features across different teams from concept to delivery
- Planning and aligning business values to the team's delivery work for objective decision-making
- Taking advantage of multiple skill sets across teams and specialists in the organization to deliver a high-quality release
- Aligning sprint goals across multiple teams
- Building system architecture and infrastructure needs into a release plan
- Using data to track progress across multiple teams, identify problems between teams, and drive towards solutions as an organization
- Understanding team and program load vs capacity for effective agile roadmapping

Logically, to scale Agile effectively, the organization needs to expand the practice in such a way that they maintain the benefits of agility while successfully addressing these and other challenges that stand in the way.

AGILE PRINCIPLES THAT SCALE WITH YOU

-  Early, frequent, and continuous delivery of valuable output (i.e. software)
-  Focus on continuous improvement through iteration and retrospective
-  Welcome changing requirements that add value, even late in development
-  Prioritize simplicity – maximize the work not done
-  Reflect the “voice of the customer” in all work and decision-making
-  Promote communication and collaboration – between all levels and members of the organization
-  Support self-organized teams

In practice, this means moving from individual Scrum or Kanban teams to coordinated efforts across multiple teams and products. It means expanding Agile practices to other teams throughout the organization beyond development so everyone is speaking the same language and working toward the same goals.

This will involve adjustments to how entire layers of the organization think about, plan, and execute their daily work. It will impact culture, collaboration, and reporting functions at all levels.

And, when it's done correctly, it will dramatically change the customer's perception of what the company offers in terms of intrinsic value.

Three Key Levels of Agile at Scale



At the **portfolio level**, the main goal is to intake, evaluate, prioritize, and track all important initiatives and ensure all activities across multiple programs are aligned with organizational strategy and goals.



At the **program level**, the main goal is to plan and prioritize the work to be performed by multiple teams working on related products or features within a strategic program. Program managers have to consider needed skills and capacity, coordinate activities, manage dependencies, and perform what-if analysis to achieve program-level goals.



At the **team level**, the main goal is to prioritize work, assign work within teams based on needed skills and capacity, coordinate activities, manage dependencies, and analyze performance to produce maximum value using available scope, resources, and time.

* Don't get too hung up on labels. As we'll see, different scaling frameworks use different terms, and you can make up your own if you want to. The concept is the important thing.

Why should you care about scaling Agile?

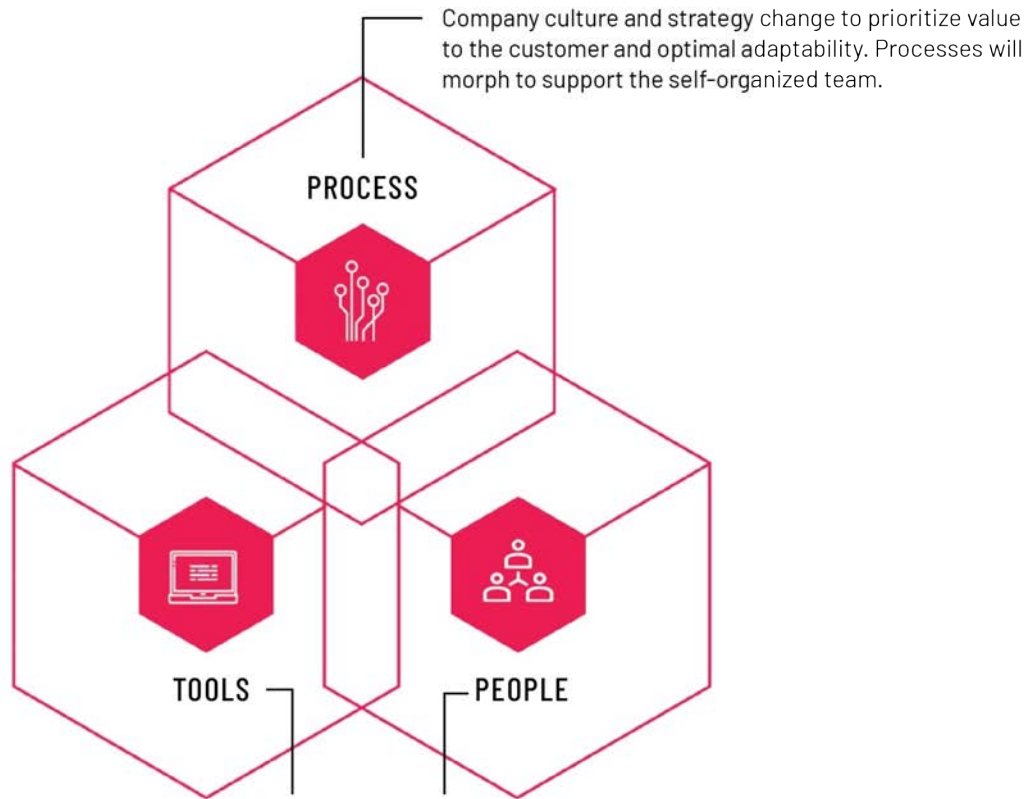
Scaling Agile at the enterprise level provides many of the same benefits to the overall organization as it does to individual teams who have embraced Agile. At the organizational level, this translates to:

Faster time to market - In a commercial world that functions at hyperspeed, the iterative release formula and emphasis on a minimum viable product (MVP) provide the speed and market visibility companies need to remain competitive. When the concept-to-cash lifecycle is shortened, everyone wins.

Driving better outcomes (i.e. value to the customer) - An Agile mindset — combined with value engineering, which we'll cover in depth in Section Three — flip the entire basis of company strategy. Rather than developing a list of features and products that check a predetermined set of boxes, the Agile organization seeks out those features and products that offer the greatest value for the customer. This enhances both short term revenue and long term customer loyalty.

Greater ROI - The iterative nature of the workflow and the value-based strategy also provide a more effective (and shorter) path to ROI, since the highest value features are always prioritized for development and they can get into the hands of the customer more quickly, requiring less initial investment.

Optimal adaptability - Change is constant and it's happening faster than ever. An Agile organization is, by definition, a nimble, innovative, and adaptable organization. In today's market, that's a requirement for long term success. In many ways, scaling Agile is about scaling adaptability.



Do you have to adhere to a particular framework for scaling Agile?

In short, no: you do not need to follow a particular framework to scale Agile effectively across your organization.

However, a framework doesn't hurt. It can serve as a sturdy inner skeleton on which the rest of your structure is built. It gives the project clear boundaries and outlines, and lends it strength and resilience. So, that's where we'll start as we begin building on this foundation.

SECTION TWO:

KEY CONCEPTS TO GUIDE YOUR BUILD

As your scaling journey begins, we'll consider the benefits and potential drawbacks of the popular scaling frameworks. Then, we'll dive deeper into two concepts that will make or break your efforts to practice agile at scale.

Pros and cons of scaled Agile frameworks

An Agile transformation of any kind involves a large number of process, team, and cultural changes, as we've already touched on. Scaling Agile at the enterprise level multiplies that complexity. That's why frameworks like SAFe, Scrum@Scale, Spotify, and others have become popular tools for organizations who have taken on that task. They offer structure for what would otherwise be a largely theoretical or abstract mission.

Scaled Agile frameworks can provide helpful reference points for getting started, an established series of signposts to refer to along your journey, and a fairly clear picture of what it's going to look like when you succeed. There are also a lot of benefits to having a common nomenclature and shared rituals that can be easily referenced over time. Of course, there's also a risk of allowing the process to take precedence over the work, in which case you're becoming less agile, not more.

In the end, it comes down to how the framework is used — not whether it's used — that matters.

The common threads running through all frameworks

While the different scaled Agile frameworks approach the process in different ways, they are actually more alike than not. That's because all of them address the same core requirements for scaling agile, and they're all based on the same basic lean and Agile principles.

For example, all the frameworks have some concept of a "team of teams" that serves to unify multiple Agile teams working on a common product or set of features. In SAFe, it's called the Agile Release Train (ART). In Spotify, it's

known as Tribes. Scrum@Scale uses the Scrum of Scrums and Meta Scrum processes to achieve the same. They all prescribe slightly different arrangements and management models, but the purpose of all these concepts is the same: to facilitate greater collaboration between teams so they can work independently or together toward a common goal.



SAFe - The Scaled Agile Framework is probably the most widely used framework to date. It is highly structured and can feel overwhelming when starting out, but it's proven its effectiveness for many global enterprises. With four distinct levels, you can choose to implement SAFe at a pace that's appropriate for your company.



LeSS - Large Scale Scrum takes a different approach from SAFe, favoring far less predetermined structure in favor of a minimalistic approach. The framework is based on all the familiar roles and rituals of single-team Scrum, simply scaled up to encompass multiple teams. It's best suited to large development operations working on one product.



Scrum@Scale - From one of the co-creators of the original Scrum, Scrum@Scale is similar to LeSS in its adherence to single-team Scrum workflows, but goes beyond LeSS with the addition of a Scrum of Scrums for Scrum Masters to coordinate with each other and a Meta Scrum for Product Owners to do the same.



Spotify - Perhaps the most unique take on a scaling framework, Spotify was created by the eponymous streaming music app maker over time as they needed to rapidly scale their own agile development organization. It's built more around the people than the process, but still retains many of the standard agile ceremonies and roles (albeit with different names.)

Other popular options that aren't technically scaling frameworks, but that have proven helpful to organizations undergoing their own scaling journey, include Lean Startup, Value Engineering, and Disciplined Agile Delivery (DA).

You can either choose to adopt a framework, which can help provide structure for your transformation, or you can create your own plan by making sure you're addressing the core requirements:

- Specialized roles with clear responsibilities
- Cross-functional agile teams and a "team of teams"
- A focus on adding value for the customer
- Integration of standard agile concepts and cadence (sprints, retrospectives, standups)
- Long-term planning and strategy with built-in adaptability
- Accounting for dependencies

- Ongoing risk management
- Coordination of development and operations at all levels (DevOps)

Teams vs Teams of Teams

The concept of a “team of teams” is fundamental to scaling Agile. It’s simply not possible to effectively develop an agile enterprise without some level of structure and coordination between different teams within the organization. Additionally, it’s important to align development teams (and teams of teams) with larger business strategies and goals at higher levels.

By establishing a formal grouping of individual teams, the organization is able to take agile practices that used to occur within each team and extend them out to the larger entity. This is really the first step in effective scaling of your agile practice from the bottom up. This accomplishes a few vital things:

- It allows for multiple teams to run on the same sprint cadence, facilitating parallel progress on related or co-dependent features.
- It allows for more effective resource management and reallocation of skills across teams as needed.
- It provides space and time for more effective long-term planning, backlog refinement, and other strategic tasks without interfering with a development team’s workflow.
- It provides a framework for effectively steering multiple teams so they’re all heading in the same strategic direction, whether they’re working on the same product or not.

We’ll exemplify the concept by using the well-known Team-Program-Portfolio arrangement, although the terminology isn’t as important as the concepts:

At the Program level, multiple development teams are brought together to work on a related set of features or products. They all follow an identical sprint schedule and they meet together periodically for large-scale planning sessions and/or retrospectives. A Program Manager (more on this role below) meets more regularly with representatives from the teams, other program managers, their Portfolio Manager, and executives as needed to guide the teams within their Program effectively.

The Portfolio level is similar in that it consists of multiple Programs whose work must be coordinated. However, at the Portfolio level, the various Programs may be running at different cadences or are otherwise disconnected from each other because a Portfolio may include many different products and business initiatives that are more loosely related to each other.

It's important to note here that scaling Agile to the enterprise level does require structure, planning, and dedicated resources that self-organizing Agile teams at the Team level don't need. In that respect, it can take some getting used to, especially for any old-school Agile purists that have a built-in mistrust of red tape. However, it's also important to note that the introduction of these necessary layers shouldn't impede development progress at all. In fact, when they're properly planned and executed, they enhance the speed, agility, and productivity of every team.

Agile Roles at Scale

As touched on above, when Agile practices are scaled from the bottom up in an enterprise, it necessitates the creation of new roles to facilitate the workflow. These aren't just layers of bureaucracy, though. These roles are specifically designed to accomplish some key tasks that will allow everyone from the Team level up to keep operating as smoothly as possible using Agile methods. Here are some of the key tasks these roles fulfill:

- Refining and prioritizing the backlogs from which teams pull their new work, ensuring every team is always focused on the next-most-important task from a customer value perspective.
- Allocating resources effectively across teams and programs so that the highest-priority work can always get finished quickly with a high level of quality.
- Working with executives and other decision makers to develop, synthesize, and effectively communicate business strategic goals to every level of the organization.

Within cross-functional Agile teams, it's common to find individuals filling three key roles:

- Someone who takes the lead in facilitating work and keeps the team on track (Scrum Master, Agile Facilitator, etc.)
- Someone who represents the customer and their needs (Product Owner)
- Someone who takes the lead in development activity (Architect)

As Agile practices scale up to the team of teams level and beyond, the same three basic roles also need to be filled at a higher level:

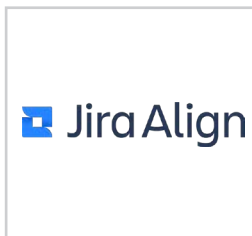
- Someone who keeps multiple teams on track through backlog refinement, resource allocation, and constant communication (Program Manager)
- Someone who represents the customer and their needs (Product Manager)
- Someone who oversees and coordinates development activity at the Program or Portfolio levels (Program/Portfolio Manager)

Two Power Tools You'll Want to Build With

These two software applications are part of the Atlassian suite of solutions, and we believe they are the best options available for effectively establishing and scaling an Agile practice in any development organization.



Jira - The Jira ticketing solution has proven to be a powerful, flexible task management system that functions perfectly as part of any Agile workflow. With hundreds of available integrations and the ability to endlessly customize the solution, Jira can easily become the “one source of truth” every organization needs to effectively scale their Agile practices.



Jira Align - Jira Align takes the awesome power of Jira and multiplies it infinitely by tying together all levels of a scaled Agile enterprise. With near real-time data flowing in from Jira as teams do the work, Jira Align’s powerful planning, monitoring, and reporting functions facilitate the work of Program Managers, Portfolio Managers, and executives at every level.

Throughout the final section of this whitepaper, we’ll be highlighting how these two power tools can help make your scaling journey easier, faster, and more successful. Not just because we work with these products every day, but because we know they’re the best tools for the job.

SECTION THREE:

A BLUEPRINT FOR SUCCESS

To round out this first volume of The Engaged Enterprise's Guide to Scaling Agile, we're going to cover some "success patterns" — tips for scaling success that we've gleaned from helping hundreds of client organizations scale their own Agile practices.

Success Pattern 1: Coordinated Planning and Execution

As noted in the last section, establishing one or more "teams of teams" with its accompanying roles is the first — and, arguably the most important — step in scaling Agile from the bottom up. The main reason these roles and organizational concepts exist is to facilitate coordination of teams and synchronization of effort. That's how the whole ends up greater than the sum of its parts.

To get the most out of this opportunity, successful organizations have found the following tips invaluable:

Unify teams on a common cadence

Get all the development teams within a team of teams unified on a common midrange cadence. This can include both sprint length — perhaps every two weeks — and longer planning increments — perhaps every three months.

Doing this makes it far easier for these teams to work independently as usual while, at the same time, functioning in tandem with all the other teams that are working on related features. When all teams are working on a common cadence, managing and planning at the team of teams level is far easier and more effective all around.

Jira Align was designed to support any team of teams structure so you can set yours up according to your unique situation. Just enter your teams as they're organized and establish your sprint start and end dates and planning increments. Then, the application pulls in all relevant data from Jira, allowing you to monitor all the work being done at the team of teams level and beyond. You can slice and dice the data any way you'd like, and you can easily review and even manipulate any backlog within the multilevel team of teams setup.

Separate team backlogs

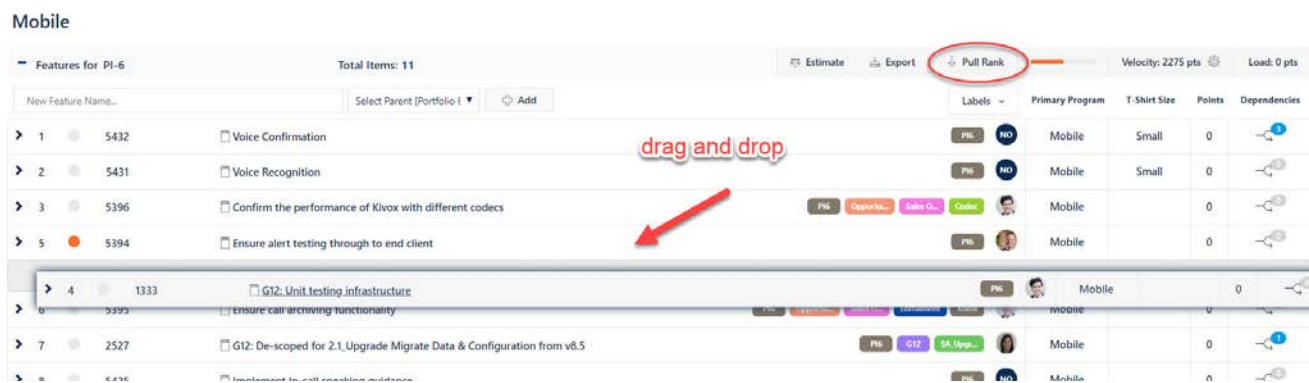
Provide each team with its own single, stack-ranked backlog of stories, but maintain an overarching backlog of features at the team of teams level. Doing so allows individual teams to work unimpeded by an extensive or cluttered list of possible tasks or stories. Instead, they can simply pull from the backlog that's already been groomed for them.

This has proven vital to successful planning and tracking of work at the team of teams level while also keeping everyone just a little more sane as they handle their day-to-day tasks.

Jira and Jira Align both support separate backlogs that can be managed behind the scenes, as well as automation through integrations that can facilitate the assigning of stories as needed. In Jira Align, all work is connected, automating much of the project management busy work, freeing up program managers to focus more on program vision and strategy.

The program backlog housed in Jira Align is an itemized list with easy drag-and-drop functionality for prioritization. Users can filter this list by program, strategic driver, owner, process step, WSJF, story points, or t-shirt sizes, for example.

And, if the strategy has changed and initiatives have been reprioritized, Jira Align users do not have to go in and manually reorder features accordingly. They can choose to “Pull Rank” to automatically rearrange features respective to the new priority assigned to the parent item.



From a reporting perspective, Jira Align provides cycle time, WIP, and throughput metrics for Kanban teams and velocity for Scrum teams to facilitate future planning. It also provides overall planning increment metrics across all teams.

Success Pattern 2: Visibility Into Everything for Everyone

Transparency and visibility are key aspects of Agile, but they don't come naturally when scaling Agile beyond the individual team level. Without a concerted effort to maintain visibility, however, effectiveness can grind to a halt. Here are some tips other companies have successfully used to keep everything visible to everyone:

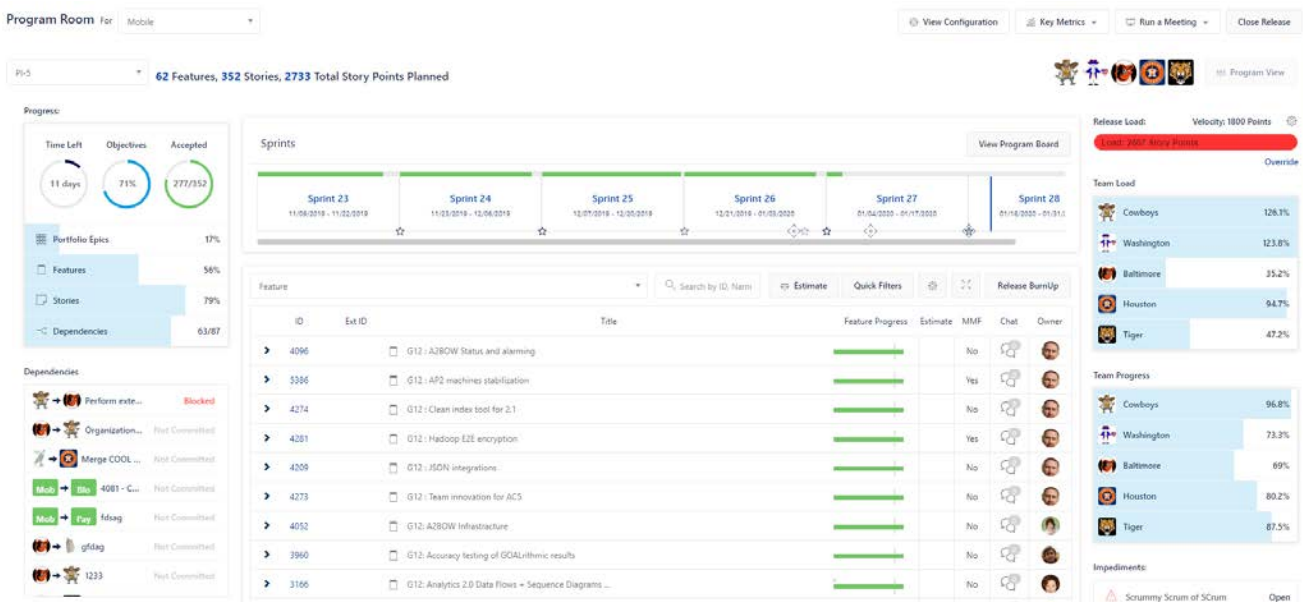
Have a single source of truth

When work is managed across numerous platforms, it's very easy for stories to slip through the cracks. At the very least, it's unproductive and annoying for team members to have to switch between applications to ensure they're getting everything done, or for other stakeholders to do the same when looking for information.

The seamless integration of Jira and Jira Align fills that need exceptionally well. Together, they provide all the functionality needed by the development teams to understand and execute quality work. Likewise, they provide all the data, reporting, and management functionality required by decision makers at the team of teams level and beyond. Any stakeholder who wants visibility into what's been done, being done, or will be done, can do so using this one unified solution.

In Jira Align, the program room provides a view of the entire program or individual team progress, team loads, risks, dependencies and much more.

Diligently manage dependencies and risks



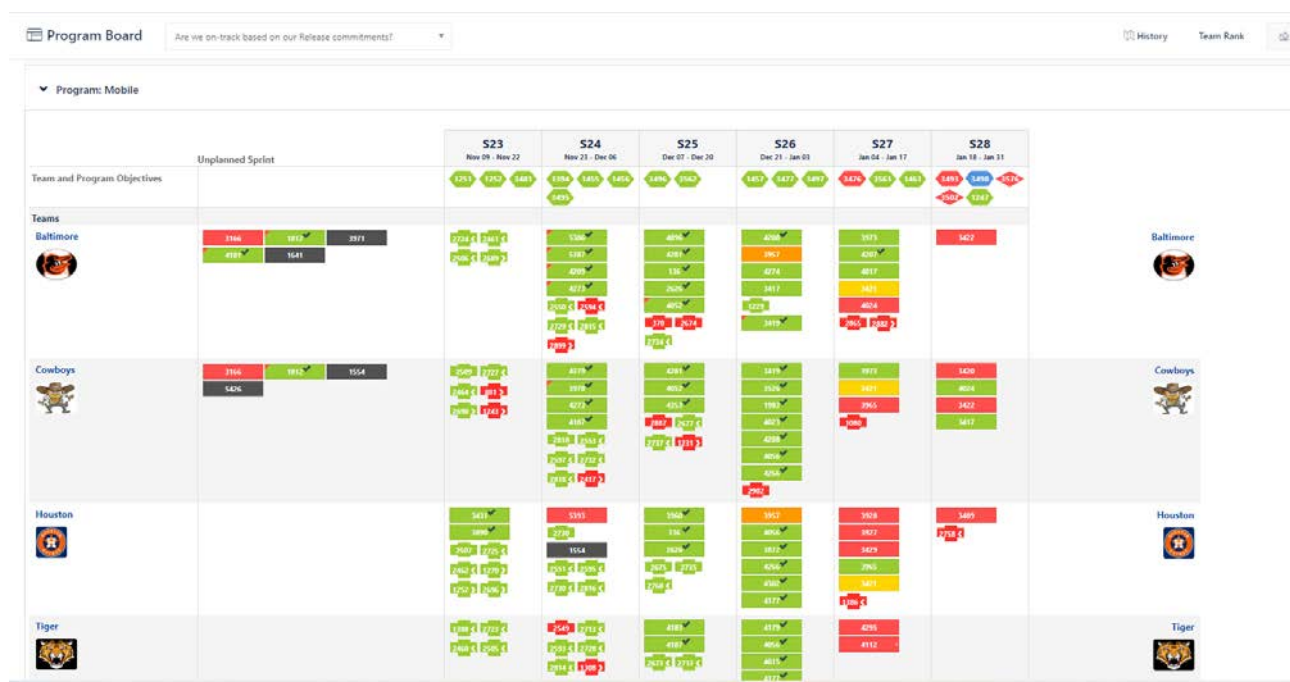
Transparency across all teams and all levels of the organization makes two of the key responsibilities of the Program Manager (or equivalent role at the team of teams level) far easier: managing dependencies and risks.

Dependencies are inevitable. Whether it's frontend development that hits a roadblock waiting on backend work to be completed, or an entire project hitting a standstill because of database schema updates, dependencies are the single biggest cause of churn across teams. Ideally, teams would be organized in such a way that dependencies would be minimized, but that's not always possible in the midst of a scaling journey.

One key to successful delivery is the tracking of dependencies with a focus on removing bottlenecks. Another is the foreseeing potential risks and taking steps to mitigate them before they impact production. Both of these vital tasks are made easier by the integrated planning and WIP monitoring functionality within Jira Align.

Specifically, the Program Board in Jira Align offers a comprehensive hub of information at the team of teams level that has proven priceless for stakeholders in numerous organizations. Serving as a digital version of the physical board program managers have been relying on for years, the Program Board provides a simple, intuitive, highly visual resource where you can gather all the information you need.

The board includes forecasting, simulation, risk tracking reports, and much more. It delivers a stunning visualization of program cross dependencies, and a deeply satisfying, comprehensive view of resource allocation across projects, with an uncanny ability to spot inefficiencies and gaps that already exist or soon will, so they can be swiftly resolved.



Jira Align Program Room

Success Pattern 3: Continuous Improvement in Planning and Forecasting

Continuous improvement through incremental delivery and feedback loops is a core element of the Agile philosophy. So, it makes sense to apply it to planning and forecasting at the team of teams level. Here are some tips successful companies have discovered as they've scaled Agile from the bottom up:

Have regular steering meetings

It's never beneficial to have meetings just for the sake of having meetings. But, some meetings obviously have more value than others. At the team of teams level, regular steering meetings can be the difference between long-term effectiveness and the erosion of value.

The Program Manager, Product Manager, and Systems Engineer should get together regularly to review the following information:

- Open risks
- Dependencies
- Feature progress
- Available resources
- Changes for the customer

The purpose of these meetings is to stay on top of risks, dependencies, and resource allocation in the midst of ongoing sprints, and to determine whether the current goals for the planning increment still make sense. If changes need to be made to ensure the teams are delivering the highest value to the customer with each release, that's going to become apparent during a steering meeting.

Once again, the Program Board in Jira Align is the ultimate tool for reference during steering meetings. It puts all the necessary data at your fingertips, and it can be customized to fit your organization's unique goals and strategy, or a specific project's unique requirements. Since Jira Align is constantly pulling in the latest data from Jira, you can be sure you're working with near real-time information whenever your steering meeting is held.

A focus on business value for the customer is vital to Agile success. As you scale agile from the bottom up, it's important not to lose sight of that focus. Here's how to make sure that doesn't happen:

Empower the Product Manager

As the customer's advocate at the team of teams level, the Product Manager needs to have full visibility into work that's in progress and work that's coming down the pipe via the backlogs. More than that, they need to have the full support of other decision makers to make necessary course corrections when customer feedback warrants it.

We already discussed the Product Manager's role in regular steering meetings for this purpose. They can and should use these opportunities to score future stories and features for value, then change priorities within the backlog and limit WIP as needed. One popular value scoring formula comes from the SAFe framework. It's called Weighted Shortest Job First ("WSJF") and it offers a way to objectively evaluate, weigh, and prioritize epics in the backlog.

The formula to calculate

$$\text{WSJF} = \text{CoD} / \text{Job Size}$$

- Cost of Delay (CoD) = Business Value (BV) + Risk Reduction (RR) + Time Criticality (TC)/Opportunity Enablement (OE)
- Job Size is the relative size of the job against the rest of the jobs/epics in the backlog. It is the first proxy for duration. Values for the above metrics are set using a Fibonacci scale.

Jira Align has a WSJF calculator built in, making value scoring quick and easy. Product Managers can take advantage of this feature repeatedly since value is a point-in-time determination. The market is constantly changing, so logically, customer value of a given story or feature can change as well.

The screenshot shows the 'Feature Estimation' interface in Jira Align. At the top, there are filters for 'Release' (PI-5) and 'Program' (Mobile). Below this, a table lists several features with their respective WSJF scores. The table has columns for Business Value, Time Value, RR/OE Value, Job Size, and WSJF Score. The WSJF Score is calculated as CoD / Job Size.

Feature	Business Value	Time Value	RR/OE Value	Job Size	WSJF Score	Points
3978 - Interface: Basic functionality with PPFW Complete basic functionality - Leftovers from IIS	1	2	20	8	2.88	100
4381 - UI: Reports in Madoop ws UI: Reports in Madoop ws	8	5	2	8	1.88	
3526 - Interface: E2E testing ITS - PPFW gfdsher	Select	Select	Select	Select	0	200
3417 - Interface: PPFW - ITS topology Testing Design assumption - performed with Prizma	Select	Select	Select	Select	0	
4181 - Interface: PPFW Descoped	Select	Select	Select	Select	0	

Jira Align Program Board

Incorporate Value Engineering into your processes

Value Engineering was highlighted earlier as an alternative some companies have successfully used to guide their scaling journey in lieu of a formal scaling framework. Whether you use a formal framework or not, however, the principles underpinning Value Engineering can be highly beneficial to any Agile development organization.

It's not easy in software development to think beyond features and really focus on the market and the customer first. Big planning cycles with slow feedback loops hinder your ability to understand your product investments. You may be making large investments in product development but then waiting a long time – sometimes on the order of years – to understand whether the investment was a good bet. There's also a good chance your company insists on defining and measuring project success in terms of outputs (such as hitting budget, time, and scope goals) rather than outcomes (like improving customer satisfaction and increasing business profitability). As a result, low value work and mediocre ideas are brought to life over long periods of time when this effort could have been put directly into high value outcomes.

That's where Value Engineering comes in. These three basic principles illustrate how to manage outcome-based activity using Value Engineering:



Hypothesize: Quantify your beliefs of what value is, and how you will know it.



Bet: Test your hypothesis with experiments to gain knowledge to make better investment decisions.



Pivot and Persevere: Learn early and often what's providing value and what's not.

Most traditional organizations drive feature lists through the portfolio and teams backlog, rarely reviewing how the work is impacting the desired outcomes of the product, and organization overall. Value Engineering makes it imperative to ensure that you frequently look at how the experiments are proceeding in terms of investment, information gathering, and value returned so that you are constantly improving your ability to understand where value occurs, and determine whether to pivot or persevere.

Jira Align has Value Engineering baked into nearly every module, and it can be utilized from the bottom up – starting at the Program level – or from the top down, or both. Jira Align also displays business value delivered versus what was planned, vital information for avid value engineers.

CONCLUSION

We covered a lot of ground in this brief whitepaper:

- What scaling Agile means and why it matters
- The most popular scaling frameworks and the value they add
- The key concepts of “bottom up” scaling, including a team of teams and Agile roles at scale
- Two powerful tools you can and should use to supercharge your scaling efforts
- Four powerful “success patterns” you can use like blueprints to engineer your own successful scaling journey

If you'd like to explore more about Jira and Jira Align, checkout the resources listed below. And, stay tuned for the release of the second whitepaper in this series – Volume Two: Scaling From the Top Down – in the near future!

