ATLASSIAN + cPRIME

Scaling agile with Atlassian and SAFe®

Dan Radigan
Senior Agile Coach
Atlassian

Brandon Huff
VP Agile Software Solutions
cPrime

Swati Jain
VP Business Process Solutions
cPrime

CONTENTS

Introduction	3
The need to scale agile	4
SAFe® overview	6
Key SAFe® roles	8
Key SAFe® activities	10
Tool requirements to support SAFe®	12
Atlassian and SAFe®	13
Atlassian configuration overview	14
Solution overview	16
Appendix	45
Team level agility	46

Introduction

As agile adoption has increased over the last decade, many organizations have grown with agile and are using scaling methodologies to help them plan, deliver, and track progress across their teams. While many scaling methodologies are available, the Scaled Agile Framework (SAFe®) has been the most widely adopted methodology by larger organizations.

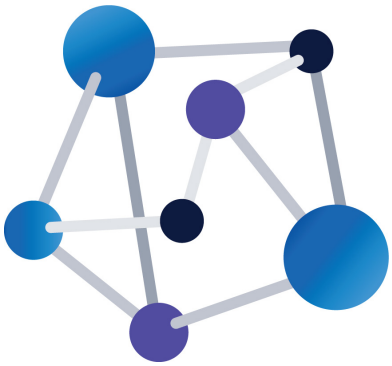
Atlassian has been supporting large companies as they adopt agile methodologies for many years. With the combined force of Portfolio for Jira and Jira Software, Atlassian provides a powerful way to scale agile. This whitepaper will discuss how Jira Software and Portfolio for Jira can support the SAFe® methodology (version 4.5) and organizational needs at all levels.

The need to scale agile

Agile methodologies (Scrum, Kanban, XP, etc.) focus on individual team planning and delivery activities. Each methodology has specific roles, ceremonies, and reports to help ensure continued delivery of incrementally valuable products. For the individual team, these approaches have a strong history of success, but until recently these methodologies have struggled to scale across multiple teams and plan work at a higher level.

As agile teams matured and grew, they became challenged with how to:

- Track large initiatives that combine multiple features across different teams from concept to delivery
- Plan and align business values to the team's delivery work for objective decision-making
- Take advantage of multiple skillsets across teams and specialists in the organization to deliver a high-quality release
- Align sprint goals across multiple teams
- Build system architecture and infrastructure needs into a release plan
- Use data to track progress across multiple teams, identify problems between teams, and drive towards solutions as an organization.



Initially, agile leaders struggled with creating a repeatable way to solve these issues. In 2011, Scaled Agile Inc. released Scaled Agile Framework for the Enterprise (SAFe®[1]) 1.0 to help address these issues and provide success patterns for large and small organizations.

Now in its latest iteration, SAFe® 4.5 has been updated based on customer feedback, application of lean principles, has the need to innovate faster, and real-world usage patterns. It also has the widest adoption as a method to scale agile.

SAFe® assumes teams are following an Agile (Scrum or Kanban) methodology. For more details about team level agility, please see the appendix.

SAFe® overview

SAFe® is an online and "freely-revealed knowledge base of proven success patterns for implementing lean-agile software and systems development at enterprise scale." SAFe® provides ceremonies, roles, metrics, and relationships that allow organizations to leverage lean and agile at enterprise scale.

This whitepaper closely aligns to the "Full SAFe®" diagram found at http://www.scaledagileframework.com/. This provides details on all levels of the methodology even if all levels would not be used at your organization. For questions on terminology or the framework, please refer to the Scaling Agile website.

We will review some of the primary SAFe® tenets and concepts here. For a quick overview of SAFe®, please view this brief video.
http://scaledagileframework.com/safe-4-0-in-5-minutes-video
While not comprehensive, the video should outline the basics to help understand the concepts and terms in this whitepaper.

The newest version of SAFe (SAFe 4.5) helps enterprises get results faster, more consistently, and more reliably.
http://www.scaledagileframework.com/whats-new-in-safe-45/
A video overview is also available: https://youtu.be/qTG4I6jUbj4

SAFe 4.5 provides additional framework, roles, events, artifacts, and explanation with a focus on:

- Validating and testing ideas faster using Lean Startup Cycle and Lean UX concepts
- A continuous delivery pipeline to align DevOps practices with planning and delivery
- Improve governance and portfolio performance using Lean Portfolio Management and Lean Budget concepts

SAFe® 4.5 focuses on practices and alignment of work at key levels of an organization:

- Portfolio level[2] – the highest level of SAFe® that provides systems and solutions to ensure the enterprise meets its strategic objectives value stream; an optional level for large and complex solutions where multiple groups (defined below) are required to deliver
- Large Solution (optional)[3] – This optional level is for organizations that need to build large-scale, complex solutions:
- Program level[4] – where team resources are applied to critical, ongoing development work
- Team level[5] – where teams work on a common iteration cadence to define, build, and test stories from their backlog

To align activities and resources, SAFe® defines and uses unique constructs:

- Agile Release Train ("ART")[6] – represents a group of 5-12 teams (50-125 people) planning, committing, and delivering business value at the program level. The ART is large enough to deliver significant business value, but small enough for everyone to have meaningful relationships with one another. It ensures a reliable schedule and fixed cadence through a dedicated team and aligned artifacts.
- Program Increment ("PI")[7] – a time box that synchronizes planning, delivery, and reviews within an ART
- Weighted Shortest Job First ("WSJF")[8] – a way to objectively evaluate, weigh, and prioritize epics in the backlog. The formula to calculate Weighted Shortest Job First (WSJF) = CoD / 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 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. See the estimation guide in the link appendix (WSJF[9]).

Key SAFe® roles

While we expect to have lean-agile leaders[10] and a community of practice[11] driving SAFe® adoption and usage, these are some of the most important roles at each level:

Portfolio level

Program portfolio management[12] – responsible for strategy, investment, governance, and program management. Primary roles are:

- Lean Portfolio Management – responsible for the highest-level of decision-making and financial accountability; drives strategy and investment funding, Agile program guidance, and Lean governance[13]
- Epic owner – responsible for identification and creation of epics aligned by vision and strategic themes. Create value statements and lightweight creative business cases. Owns portfolio backlog, WSJF prioritization, and go/no-go decisions for completed work[14]
- Enterprise architect[15] – responsible for ensuring enterprise-wide architectural system, platform, and infrastructure dependent work is identified and created

Large Solution level

- Solution Train engineer[16] – Responsible for facilitating large solution processes and execution
- Solution management[17] – Responsible for large solution content, artifacts, requirements, and documentation
- Solution architect/engineer[18] – Responsible for alignment with enterprise architecture, as well as identifying and creating solution architecture
- Customer[19] – Internal or external resource responsible for planning, evaluating solution increments, review status of work, and provide testing/UAT feedback

Program level

- Release train engineer[20] – Responsible for facilitating value stream and ART processes and execution including PI Planning, alignment with vision, and value stream objectives
- System architect/engineer[21] – Responsible for alignment with enterprise and solution architecture, and identifying and creation of solution architecture to be delivered by teams architecture
- Product management[22] – Responsible for product backlog content and prioritization
- Business owner – Responsible for fiduciary, governance, efficacy, and return on investment for an agile release train[23]

Team level

- Product owner[24] – Responsible for defining stories, prioritizing the team backlog, review, and accepting stories
- Scrum master[25] – Responsible for lean-agile leadership, agile process facilitation, enabling the team, and removal of impediments
- Scrum team[26] – Group of individuals responsible for defining, building, and testing components/features within their agile process

Key SAFe® activities

Each level will have specific ceremonies or activities to ensure appropriate inputs and outputs as well as a sustainable process that builds in quality. These include:

Portfolio level

- Create strategic themes[27] – Identify, analyze, and create strategic business goals and objectives
- Define value streams[28] – Identify and create the value streams for your organization
- Define portfolio backlog – Create epics aligned to vision and themes and objectively prioritized using WSJF

Large Solution level

- Define Solution Intent – Define the key solution behaviors and requirements that can be used to validate and verify the delivered solution
- Define Large Solution backlog – Create epics and capabilities aligned to solution intent and objectively prioritized using WSJF
- Pre- & Post- PI Planning – Provide overall solution context, objectives, and milestones as inputs for effective PI Planning sessions
- Solution demo – At the end of each PI, each ART team will provide an integrated demonstration of the capabilities completed in the last PI
- Inspect and Adapt (I&A) – At the end of each PI, review planning and delivery processes to identify an actionable backlog to continuously improve in future PIs

Program level

- Form and launch ARTs[29] – Organize which team members, skillsets, and which product/service/feature a release train will deliver
- PI planning[30] – At the beginning of each PI, teams will plan the next PI by estimating work and identifying dependencies
- ART sync (scrum of scrums and PO sync)[31] – Alignment meetings to ensure teams and POs are aligned on work progress, challenges, and next steps
- System demo[32] – At the end of each PI, teams will demonstrate the progress on the solution to key stakeholders
- Inspect and Adapt (I&A) - At the end of each PI, review planning and delivery processes to identify an actionable backlog to continuously improve in future PI

Team level

- Iteration planning[33] – Before each iteration, teams and POs will organize their work and define a realistic scope for the upcoming iteration
- Iteration execution[34] – Delivery, impediment resolution, and tracking of iteration progress
- System demo[35] – At the end of each iteration, teams will demonstrate the progress on the solution to key stakeholders
- Iteration retrospective[36] – A candid team review of what worked, what didn't, and what should be done differently each iteration

In future sections, we will demonstrate how to use the Atlassian suite to support the SAFe® 4.5 methodology with these key SAFe® roles and activities.

Tool requirements to support SAFe®

Scaling methodologies have stretched many organizations' tools to their limits as the teams adopt the processes and practices required. Tools that succeed with scaling methodologies like SAFe® provide functionality around flexible configuration and usage at all levels. We have found the following elements are critical in a tool that supports SAFe®:

- Support for SAFe® ceremonies and activities – Ability to support multiple portfolio and program kanban boards, performance objective evaluation, and facilitate all planning, execution and demo activities
- Flexible requirement hierarchy – Ability to allow multiple levels of nested requirements
- Traceability of requirements – Ability to trace all requirements to core business objectives and themes
- Enable communication – Ability to communicate at a macro and micro level within the tool
- Support collaboration – Ability to enable cross-team and cross-ART collaboration with tracking of changes and updates
- Tracking and reporting – Ability to report on progress and challenges at the portfolio, program, and team level; provide real-time visibility to all interested parties

Tools with these capabilities will make adoption of SAFe® easier, more flexible, and provide value to all teams.

Atlassian and SAFe®

Atlassian was founded in 2002 with a mission to unleash the potential in every team. Atlassian's suite of team collaboration software – Jira Software, Confluence, Bitbucket, Hipchat Data Center, Jira Service Desk, and Portfolio for Jira – removes the friction inherent in teamwork, making it easier for teams to organize, discuss, and complete work.

Today, Atlassian's products serve teams of all shapes and sizes in virtually every industry – from software and technical teams to IT and service teams; from sales and marketing teams to HR, finance and legal teams. The engineering investments Atlassian has made in ensuring these products support every team reflects the continued commitment to R&D.

Atlassian's Jira Software, Portfolio for Jira, Confluence, and Hipchat Data Center are the foundational products required to support a SAFe® solution. Each product plays a special role in providing the flexibility and functionality required to support SAFe®.

- Jira Software is the #1 software development tool used by agile teams with customizable requirement types, workflows, permissions, and notifications. It provides virtual scrum and kanban boards for teams to collaboratively manage backlogs, deliver work, and use real-time agile reports.
- Portfolio for Jira provides a centralized interface for managing cross-team requirements, resources, and schedules with a customizable hierarchy
- Confluence is the place where agile teams create plans, discuss options, and record decisions
- Hipchat is the group chat app designed to help teams communicate in real-time

Atlassian configuration overview

There are many ways to interpret and adopt a SAFe® methodology and each organization must choose which elements are critical to them and how to leverage the proven success patterns provided. The solution discussed below will show one approach to supporting SAFe® using Atlassian products.

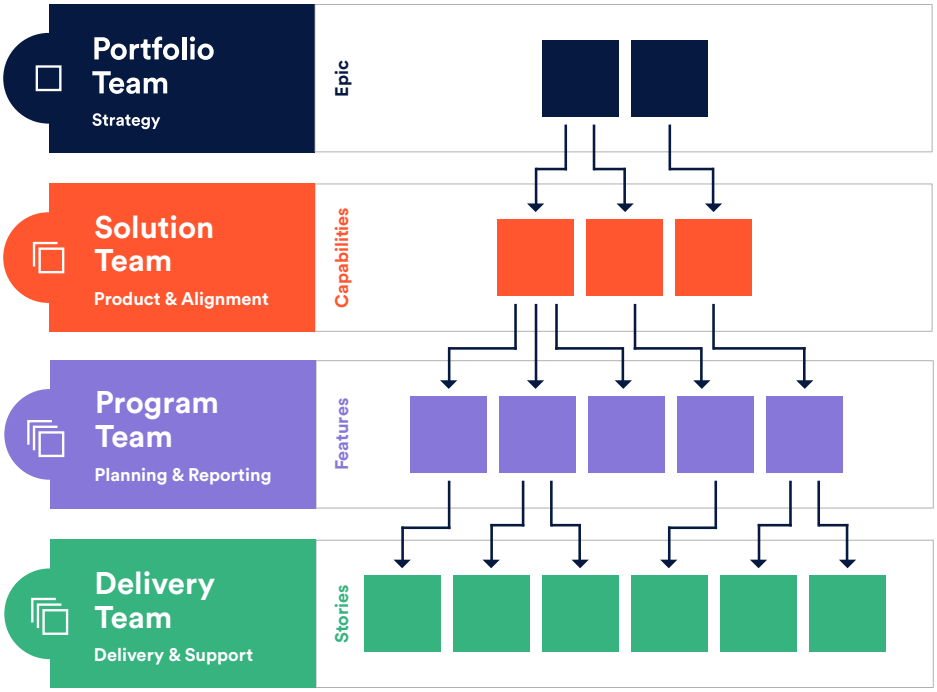
We recognize that each organization is structured differently and this solution provides one approach. Regardless of the specific org structure and roles within it, we find a common theme in how ideas, requirements, and communication flow through the tool and process.

Each level in an organization would drive key practices, activities, and deliverables in SAFe 4.5:

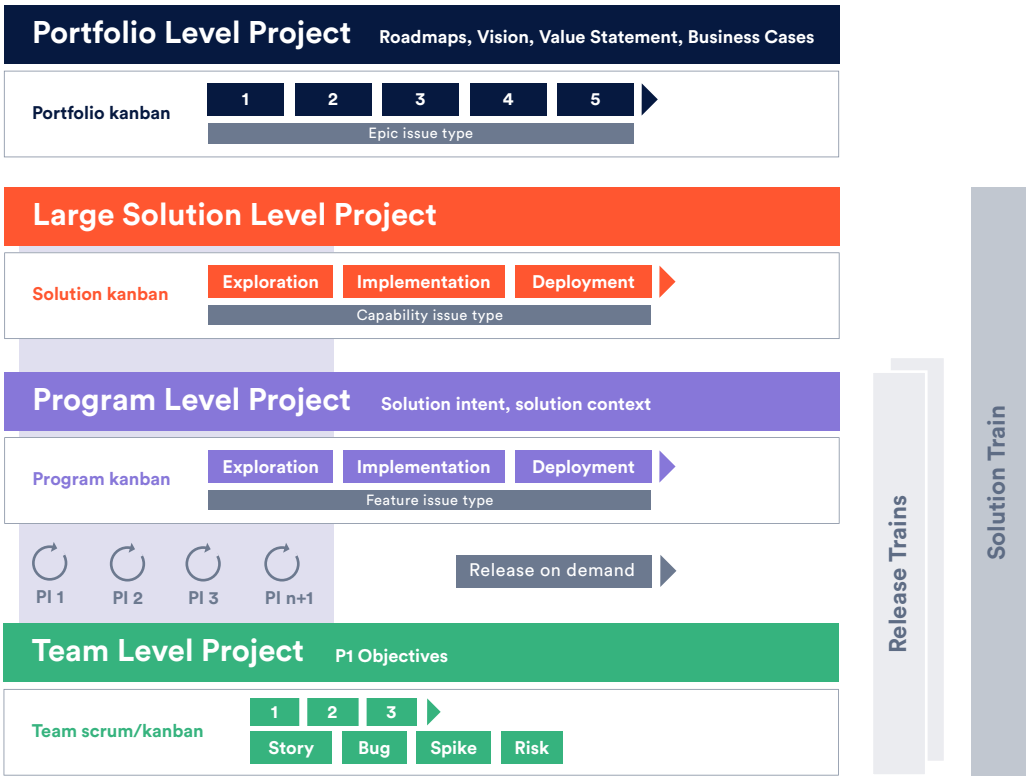
- Portfolio level – The portfolio level focuses on idea generation that is captured through an intake funnel. These ideas are vetted for economic and business viability; the approved epics are evaluated at the Large Solution and program levels.
- Large Solution level – The optional large solution level focuses on aligning the direction of large and complex solutions across multiple ARTs. This level facilitates the activities and artifacts needed to capture functional and non-functional requirements in solution intent to ensure compliance with regulations and standards. Capabilities are created to encompass solution deliverables with children features.
- Program and product level – Your program teams break those initiatives into features, establishes dependencies, estimates, and bundles these into program Increments or releases. These features maps to the team's backlog so they are in alignment for delivery.
- Team level – Teams are focused on development and delivery of planned feature functionality and track these items and their rollup to the higher level features and capabilities.

Let's take a closer look at the primary needs of each level and how they can accomplish these goals using the proposed tools.

Scaling agile Atlassian solution - hierarchy



Scaling agile Atlassian solution - structure and configuration



Portfolio level

At the portfolio level, the main goal is to intake, evaluate, prioritize, and track all important initiatives. Here are the key activities performed at the portfolio level:

EPIC OWNERS

Activities

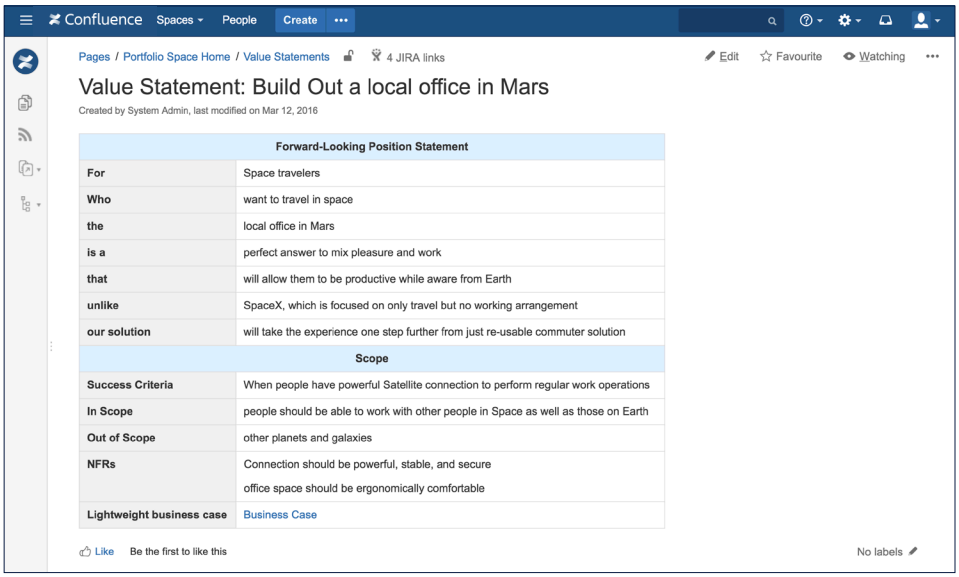
The Epic owners submit value statements and lightweight business cases

Tools used

Confluence for template-based documentation

Configuration

Portfolio space with business case, value statements, value streams templates

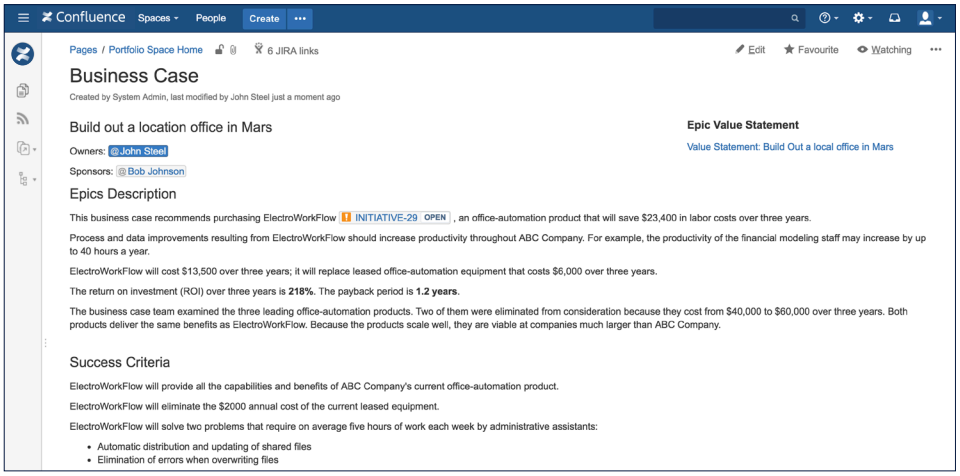


Forward-Looking Position Statement	
For	Space travelers
Who	want to travel in space
the	local office in Mars
is a	perfect answer to mix pleasure and work
that	will allow them to be productive while aware from Earth
unlike	SpaceX, which is focused on only travel but no working arrangement
our solution	will take the experience one step further from just re-usable commuter solution
Scope	
Success Criteria	When people have powerful Satellite connection to perform regular work operations
In Scope	people should be able to work with other people in Space as well as those on Earth
Out of Scope	other planets and galaxies
NFRs	Connection should be powerful, stable, and secure office space should be ergonomically comfortable
Lightweight business case	Business Case

Value statement in Confluence

Business case in Confluence

Note - Confluence allows you to create an issue in Jira



EPIC OWNERS

Activities

The Epic owners submit their requests to the funnel for further review and approval

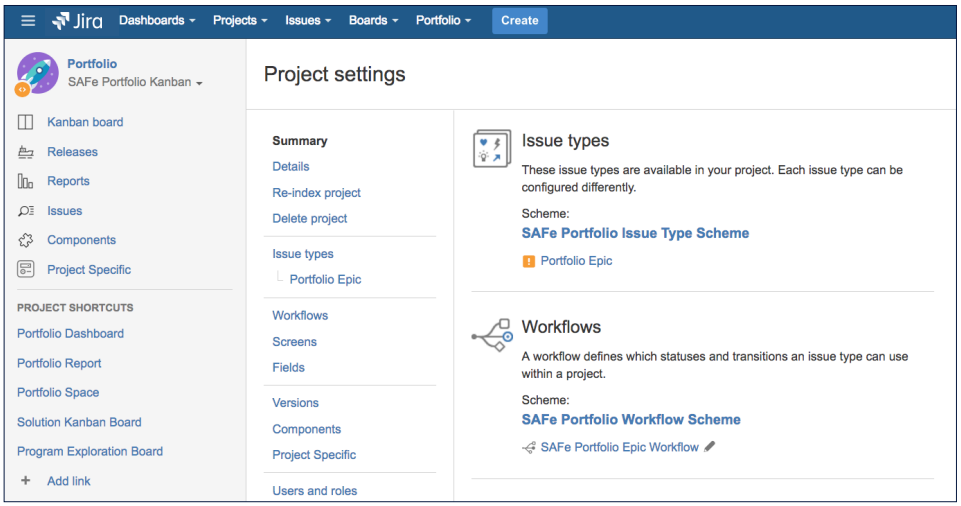
Tools used

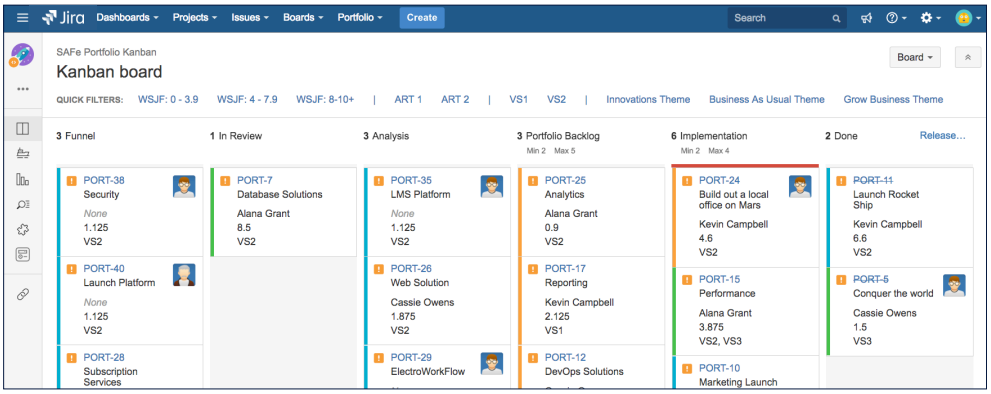
Jira Software kanban board for intake and flow management

Configuration

- One portfolio level JIRA project with a custom Epic issue type
  Note - this is a custom issue type to match the SAFe hierarchy and is not the same as a Jira Epic
- Portfolio level Jira Software kanban board based on portfolio level project and Epic workflow

Portfolio level project





Portfolio level kanban board

LEAN PORTFOLIO MANAGEMENT

Activities

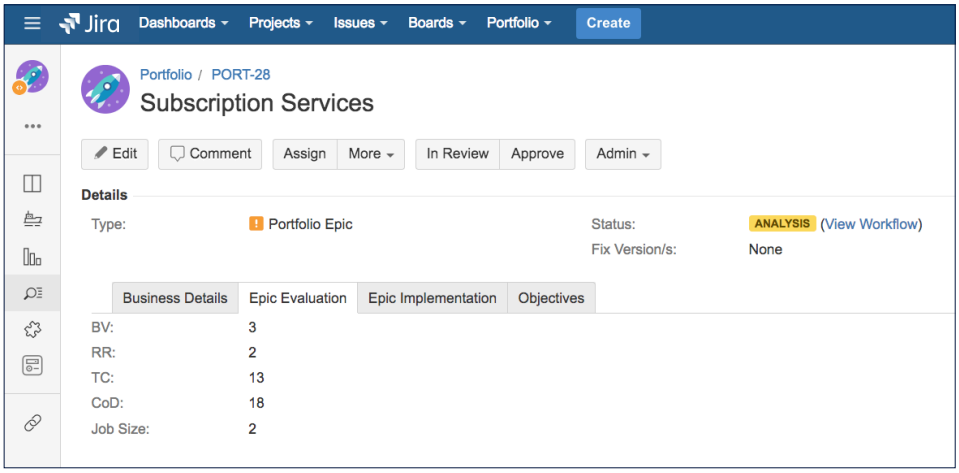
The Lean Portfolio Management team provides objective guidance to members of the Portfolio team to assist with evaluation and alignment of the epic with strategic themes and vision

Tools used

Jira Software custom calculated fields for WSJF calculation using *Power Scripts*, Portfolio for Jira for visibility and strategic alignment

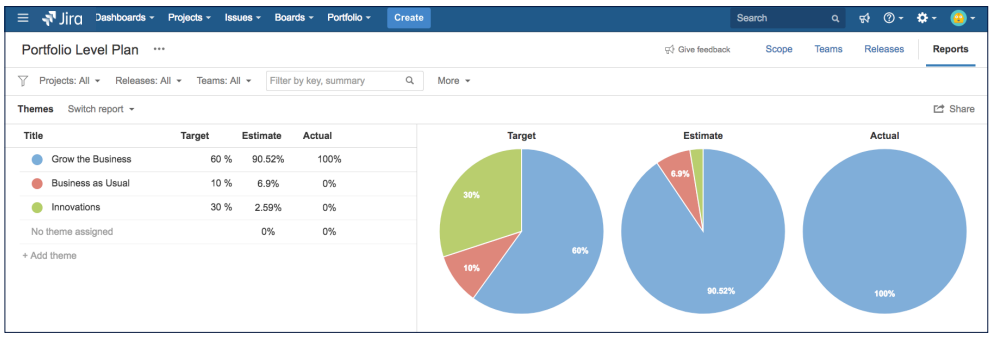
Configuration

- Configure Jira Software custom fields for intake measures such as BV, RR, TC, CoD, Job Size, and WSJF
- Configure WSJF as a calculated field using *Power Scripts* to compute the score based on the input of the other custom fields
- Manage Epic priority and ranking on the kanban board using the calculated WSJF

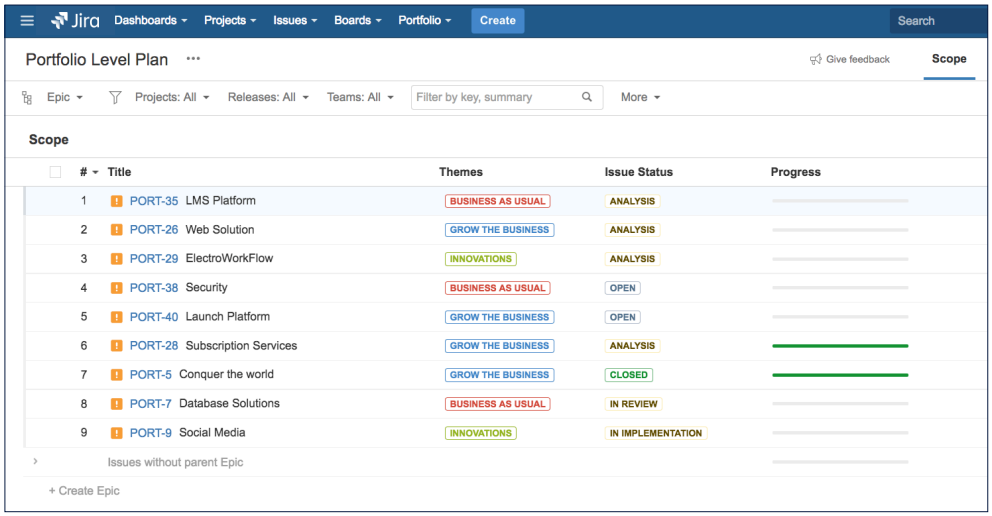


Calculated WSJF using Power Scripts

Portfolio for Jira can be leveraged to view the Epics and assign themes to ensure alignment of work. A Portfolio level board is created using the Portfolio kanban board. Only the scope view is used for this to enter existing work at this level to identify themes, their allocation, and Epic assignment to them.



Calculated WSJF using Power Scripts



Epic ranking and theme assignment in Portfolio for Jira

PORTFOLIO LEVEL ASSUMPTIONS

- It is assumed that portfolio or value stream funding is in place and funds are allocated by PPM towards epics using the guidelines of lean agile budgeting.
- The configuration approach highlights how Atlassian can be used as a solution for one value stream. The same logic can be adjusted and repeated for additional/multiple value streams.
- SAFe® recommends portfolio level epics and enablers and issue type naming has been simplified to use epics for the purpose of this solution. An organization may choose to follow any other naming conventions suitable to their needs.
  NOTE – the portfolio level epic is not the Jira Epic with layers enabled under it.
- Jira Software configuration has been simplified for the purpose of this solution. Each Jira Software project can be further configured to have its own workflow, field, screen, permission and notification schemes to align with enterprise needs.
- Confluence configuration has been simplified for the purpose of this solution. Confluence can further be configured to have tailored templates to align with organizational needs and documentation standards.

Large Solution level

At the Large Solution Level, the main goal is to align and streamline solution approach across multiple ARTs to deliver a large complex solution. While this level is optional, it is the solution for enterprises that face the biggest challenge of building large-scale solutions. This level also serves as a good anchor for governance practices and standards that may be internal or external to the enterprise. Here are the key activities performed at the Large Solution level:

SOLUTION ARCHITECT/ENGINEER

Activities

Once solution Capabilities are prioritized, the Solution Architects/Engineers will establish a common vision for the functional, technical, and architectural components of the solution. They will assess, analyze, and design the solution in a manner that bridges the gap between the current-state and future-state in the most optimal manner. This group will also perform feasibility analysis of technical assumptions and explore alternate options and architect the solution with a system mindset. Activities include:

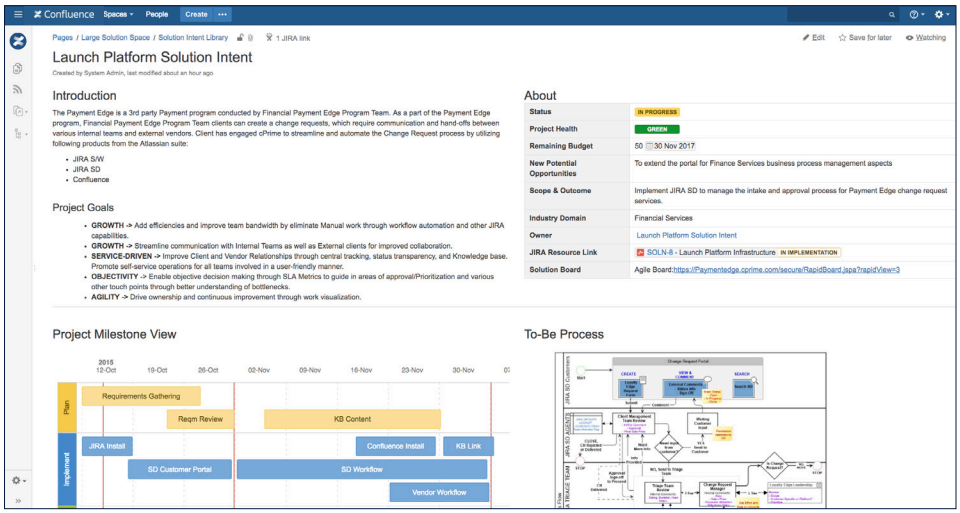
- Establish a high-level solution intent with solution models for a visual representation of how all components and sub-components interact together in the future state
- Elicit functional requirements, process flows, data flows in a comprehensive and simplified manner
- Capture non-functional requirements (NFRs)
- Pre- and post-PI planning sessions, solution demos, and inspect and adapt sessions

Tools used

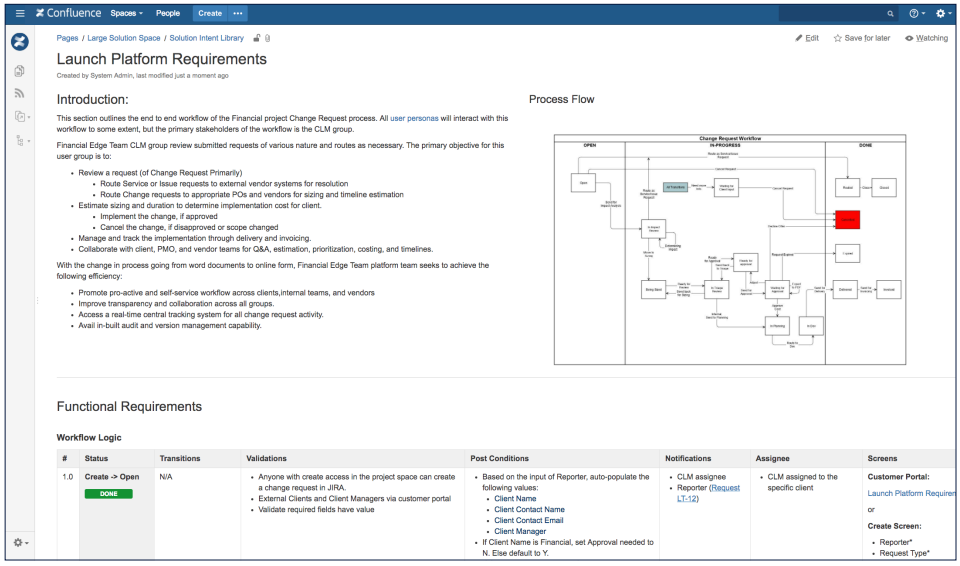
- Confluence for solution intent artifacts and for other PI related documentation

Configuration

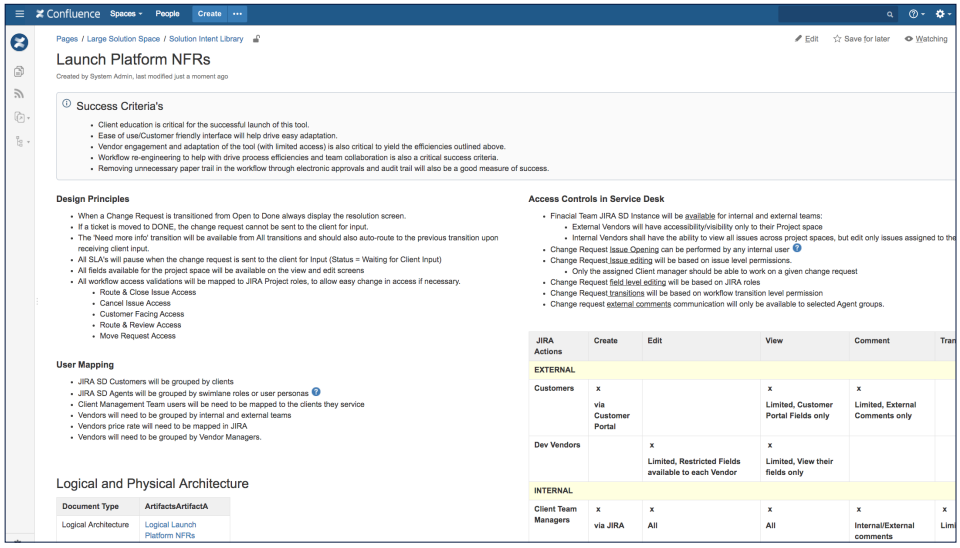
- One large solution level Confluence space per value stream
- One page branch for each portfolio level epic with child pages each for solution intent, functional requirements, NFRs, PI Planning Notes, solution demo notes, and other key documents
- Space Templates for each type of content along with pre-set page labels
- Page Libraries for each content type pulling and organizing content dynamically based on labels



Solution intent in Confluence



Functional requirements in Confluence



Non-functional requirements in Confluence

Note - Confluence offers powerful macros to tag, search, and organize content by one or multiple labels to improve contextual "findability" of your content and build dynamic content libraries.

SOLUTION MANAGEMENT

Activities

Once Portfolio epics are prioritized, the Epic Owners will facilitate the process of decomposing epics into capabilities and features with input from Solution architects/management and Product Managers. The Solution architects will review both the Kanban workflow and relationship between epics, capabilities, and features. Activities include:

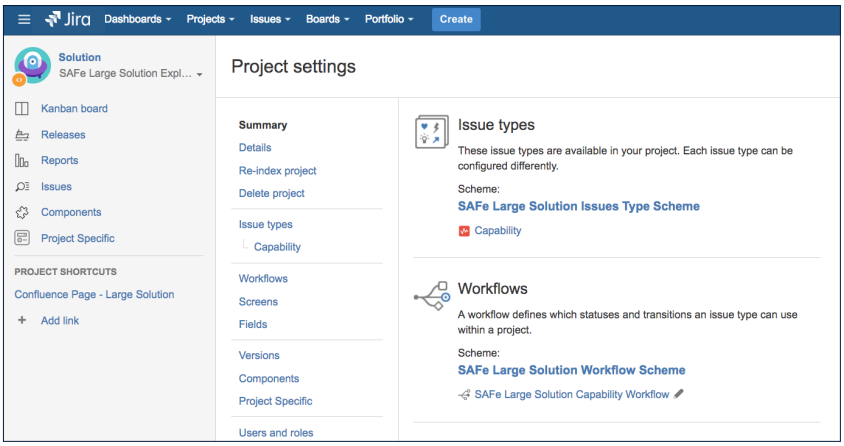
- Create capabilities where needed to align work to epic vision and themes
- Create hierarchy of epics, capabilities, and features
- Breakdown epics into capabilities and features
- Commit information with Jira Software

Tools used

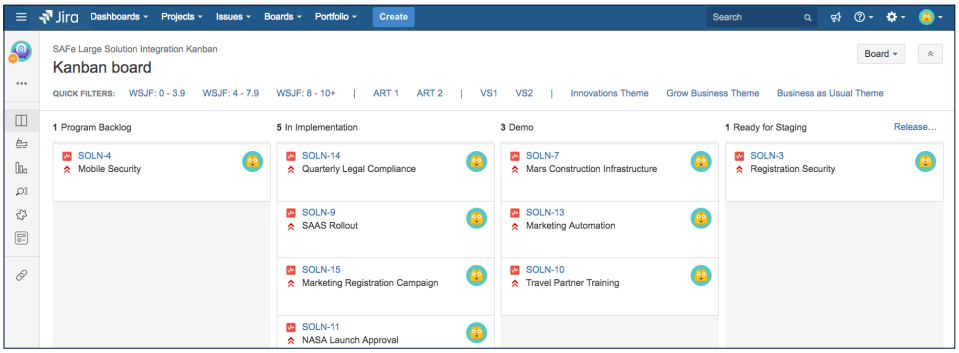
- Jira Software kanban board for intake and flow management
- Portfolio for Jira for epic and capability breakdown

Configuration

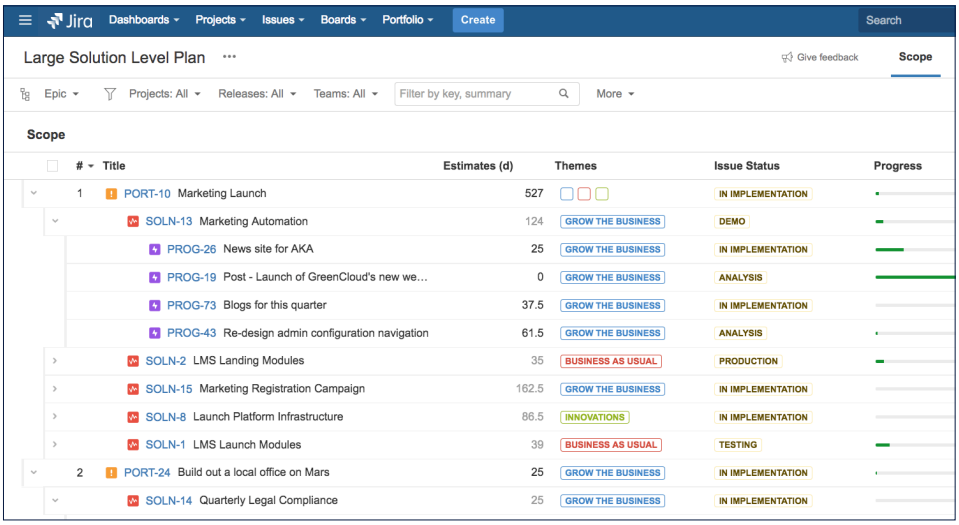
- One large solution level Jira project with a custom capability issue type and workflow that encompasses the continuous delivery pipeline
- One Portfolio for Jira Large Solution level plan with portfolio, large solution, and program projects (no boards or filters) *NOTE: Schedule and team capacity functionality will be leveraged at the program level to align with pre- and post-PI Planning deliverables. While this can be done at the Large Solution level, we have consolidated the PI work into the Program level.*



Large solution level project



Large solution level kanban board



Create hierarchy of work in Porfolio for Jira

LARGE SOLUTION LEVEL ASSUMPTIONS

- The solution highlights how Atlassian can be used as a solution for one value stream. The same logic can be adjusted and repeated for additional/multiple value streams.
- SAFe® recommends capability and enablers at the large solution level. Issue type naming has been simplified to use capabilities for the purpose of this configuration. An organization may choose to follow any other naming conventions suitable to their needs.
- Jira Software configuration has been simplified for the purpose of this configuration. Each Jira Software project can be further configured to have its own workflow, field, screen, permission and notification schemes to align with enterprise needs.
- Confluence configuration has been simplified for the purpose of this solution. Confluence can further be configured to have tailored templates to align with organizational needs and documentation standards.

Program level

At the program level, the main goal is to prioritize work, assign work across teams based on needed skills and capacity, coordinate activities, manage dependencies, and perform what-if analysis for optimal throughput using available scope, resources, and time. Here are the key activities performed at the program level:

PROGRAM MANAGEMENT TEAM

Activities

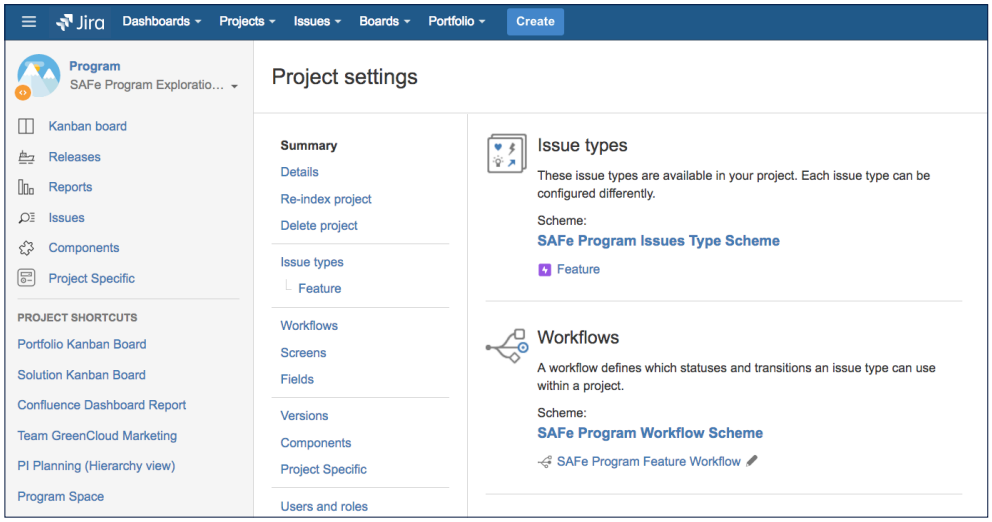
The Release Train Engineers, System Architects, and Product Management team reviews new features, manages demand and continuous flow of value using the program level kanban board

Tools used

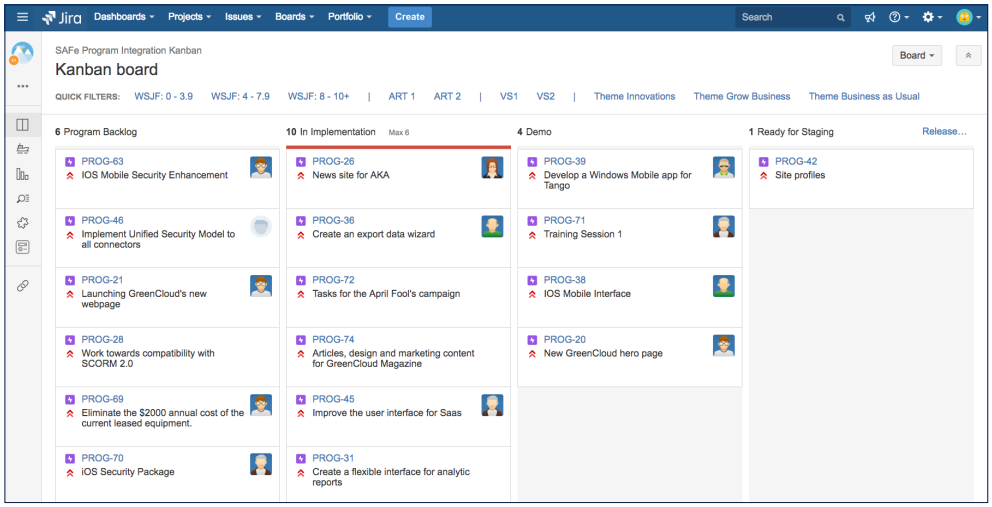
- Jira Software

Configuration

- Create a Jira Software project space for program level features. *NOTE – Jira epics are renamed to feature*
- Create a corresponding JIRA Software kanban board to visually track and manage the flow of your features
- A customized workflow allows features to move between Exploration, Implementation, and Deployment boards to manage work and focus on each aspect of the continuous delivery pipeline



Program level project

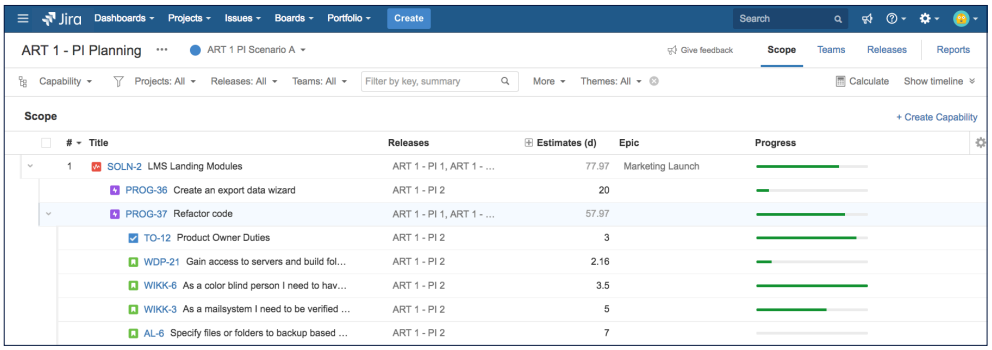


Program level kanban board

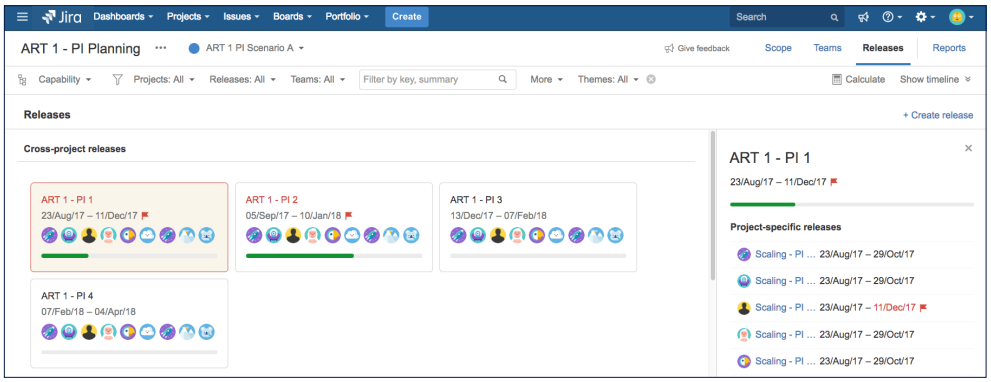
PRODUCT MANAGEMENT

Activities	Capabilities (if needed) and features for break down of work to support PI planning. The Product Manager may work with System Architects, Business Owners, Product Owners and key Team Members to break down work which would be reflected in Portfolio for Jira.
Tools used	Portfolio for Jira for PI planning and breakdown
Configuration	Create a plan for work breakdown to support PI planning encompassing the large solution (if needed), program, and team levels • Scope of plan includes all team boards (refer to team level below) • Scope of plan includes items of all capabilities (if needed) and features in Implementation workflow status. Exclude work that was not initially planned at the portfolio level • Create shared or private scrum teams with relevant team members, default velocity, estimated capacity per team member and map to each of the respective scrum board

Breaking down work in Portfolio for Jira to support PI planning



Align work by cross-project release PI in Portfolio for Jira



PRODUCT MANAGERS, RELEASE TRAIN ENGINEER, & TEAM

Activities

The PI Planning team evaluates work and performs what-if analysis to help maximize scope, time, and capacity for each PI.

The PI Planning team adds capabilities (if needed), features, and stories to a cross-project release to plan the work and determine what is feasible to deliver in this CPR. The teams will use Portfolio for Jira to view the schedule, adjust planned items, evaluate changes, and determine what work can be accomplished in a PI.

For features mapped to CPR for the current PI plan, the Team can determine which backlog items can fit into the PI based on backlog priorities, work estimates, and team capacity. Since the CPR is aligned to a PI, the schedule can show if you a PI is overbooked and turn red indicating the PI release date will not be met.

To fix an overbooked PI, the team can add more capacity, move the release date, or reduce scope.

A PI may have dependencies that need to be addressed prior to starting work. Using the dependency report, the team can view dependencies at any level to ensure they are discussed and mitigated.

While managing the scope of a PI, the PI Planning team can evaluate a number of alternative PI plans using scenarios to make changes to the underlying assumptions around prioritization, team capacity, and estimates. For each potential change, the team can view the impact using the calculate functionality.

Tools used

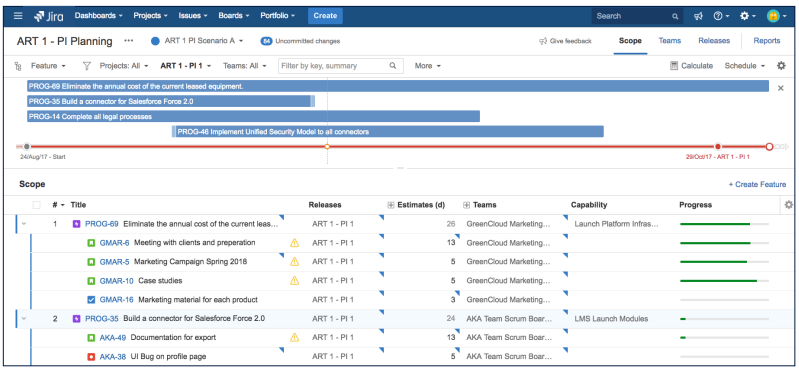
Portfolio for Jira for PI planning and additional breakdown or work

Configuration

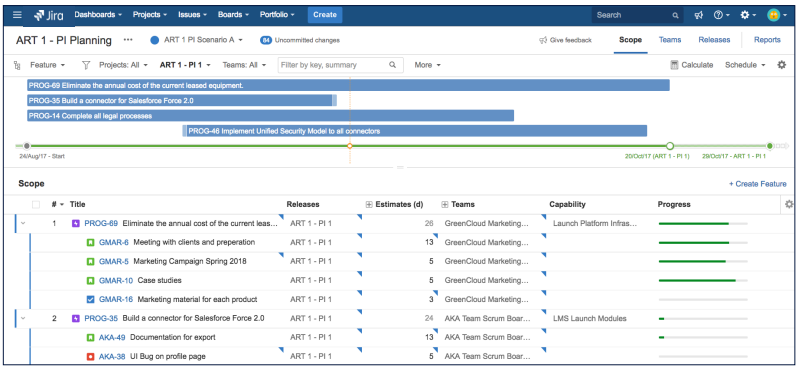
No additional configuration required

Use existing Portfolio for Jira Program level plan

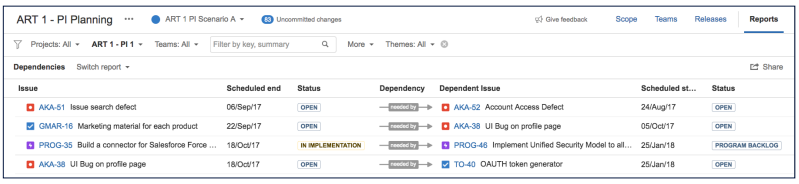
PI overbooked based on scope of work in the CPR for PI 1 in Portfolio for Jira



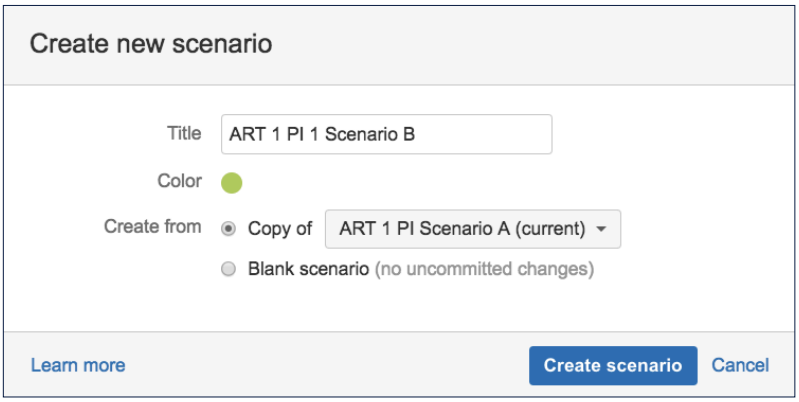
Overbooked PI 1 schedule corrected in Portfolio for Jira



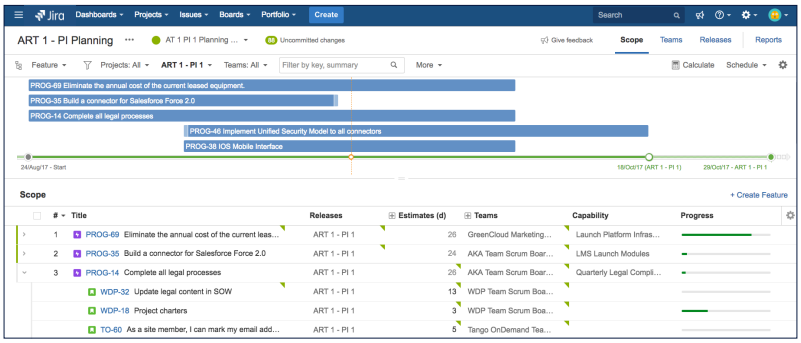
View dependencies in Portfolio for Jira by level, release, team project and more



Scenarios in Portfolio for Jira allow a team to evaluate different constraints such as ranking, estimates, team capacity and more



Scenario created to evaluate other options for PI 1 to optimize throughput



RELEASE TRAIN ENGINEER

Activities

Once prioritized, the Release Train Engineer may move the plan to execution

The Release Train Engineer commits the work in the plan which makes these changes to the corresponding issues, fields, and fix versions in Jira. This synchronization uses the plan as the point of record and ensures the correct data is available for the teams to deliver work.

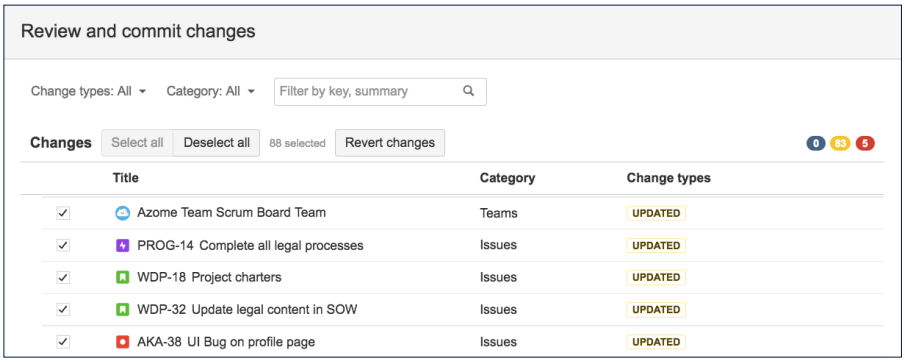
Tools used

Portfolio for Jira to promote centralized plan while trusting the team to make appropriate decentralized decision-making

Configuration

No additional configuration required

- Use existing Portfolio for Jira program level plan



Committing to planned work in Portfolio for Jira

RELEASE TRAIN ENGINEER

Activities

Release Train Engineer, Product Manager, System Engineering, and optional Business Owners to collaborate on PI and sprint planning items

Tools used

Hipchat

Configuration

Hipchat rooms configured for key roles and activities:

- Release Train Engineers

- Product Management
- System Engineering
- PI Planning (all program levels roles)
- Each Agile Team in the ART

RELEASE TRAIN ENGINEER

Activities

Release Train Engineer may have a PI page that includes the scope, meeting notes, risks registers, PI objectives and progress report of the current PI. This is viewed by Product Managers, System Engineering, and optional Business Owners, and teams, to ensure they are all aligned.

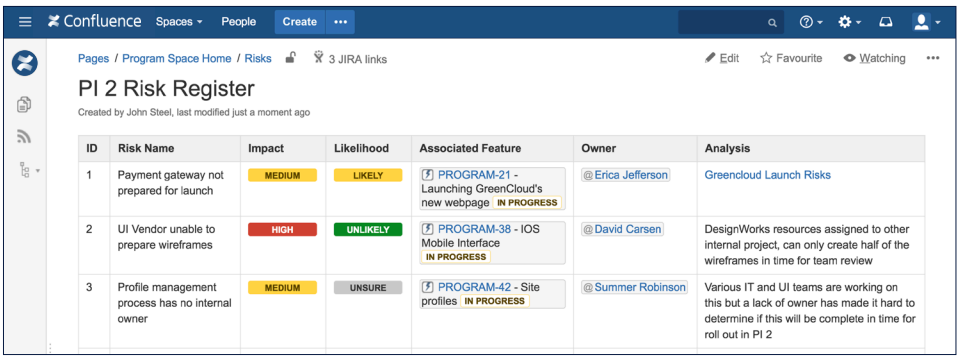
Tools used

Confluence

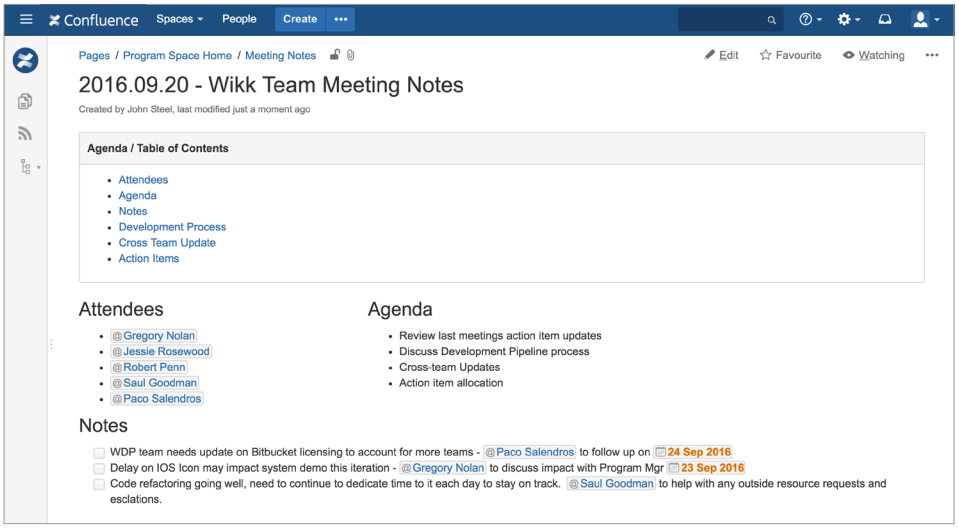
Configuration

Program level Confluence capturing key PI documentation:

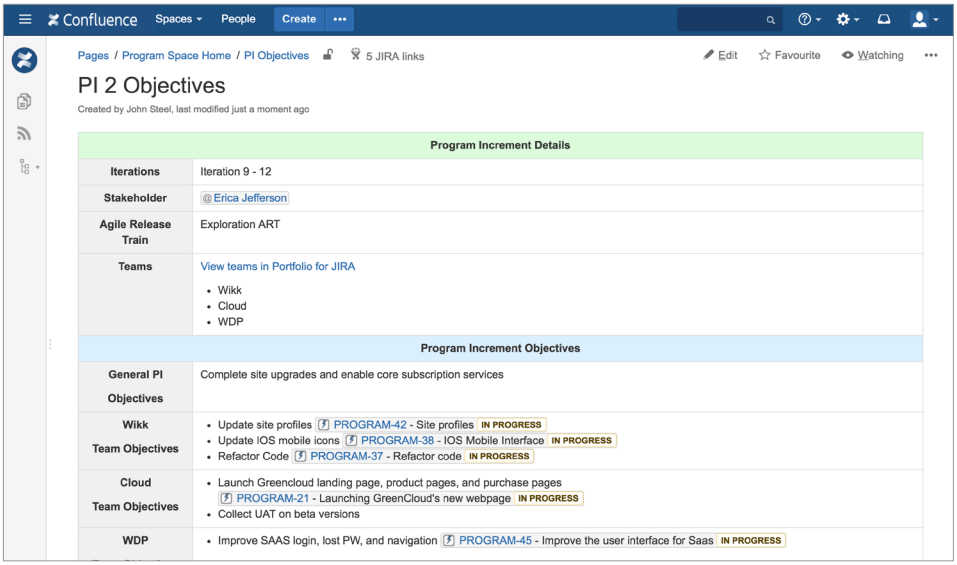
- Risk register template
- Meeting notes template
- PI objectives template
- Task lists
- Jira Software gadgets for reporting progress for a release
- Use HTML macro to include embedded links from Portfolio for Jira into Confluence page



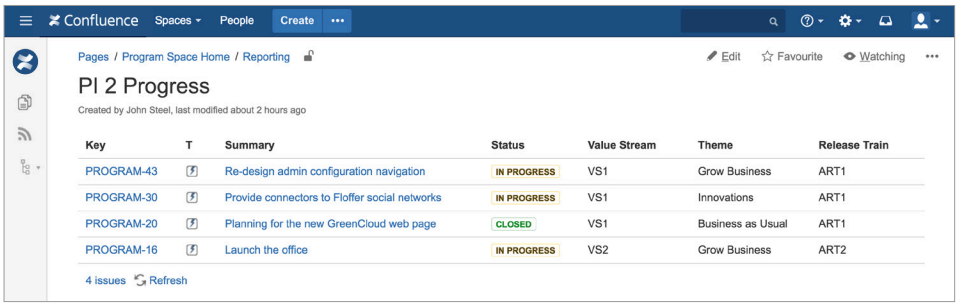
Page view of risk register in Confluence



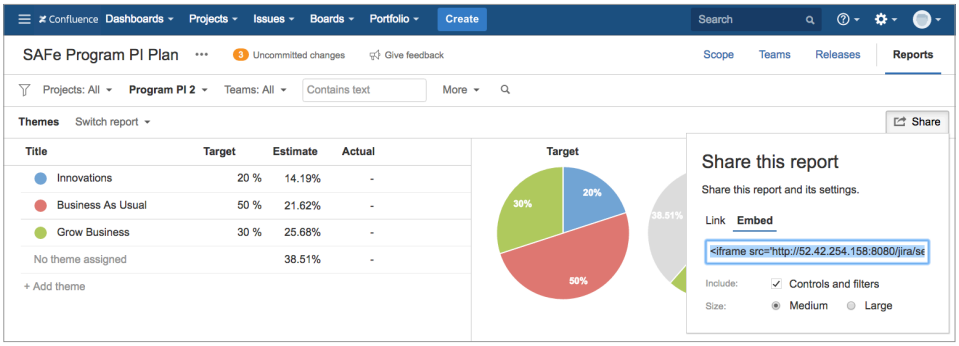
Page view of meeting notes in Confluence



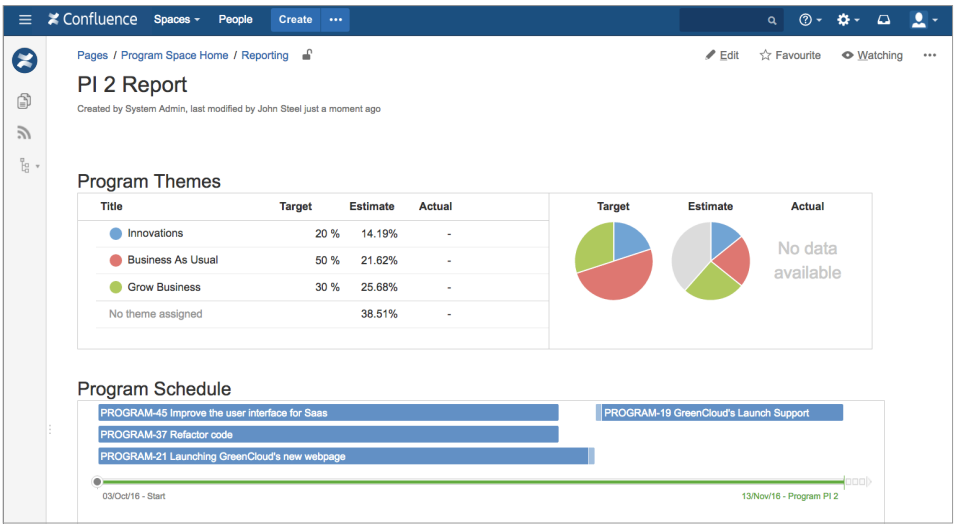
Page view of PI objectives in Confluence



Status report in Confluence



Portfolio for Jira report embedded in Confluence



Multiple Portfolio for Jira reports embedded in Confluence

Program Level Assumptions:

- It is assumed that requirements at the program level have already been through evaluation and prioritization process for it to show up in a specific PI plan
- This solution highlights how Atlassian can be used as a solution for multiple teams in one PI and one ART. The same logic can be repeated for additional/multiple ARTs.
- The solution assumes that all teams are operating as scrum teams. You may have scrum and kanban teams as part of your ART and this solution can leverage Jira Software boards for each need. Portfolio for Jira will allow you to have mixed methodology teams.

- SAFe® recommends program epics, program enablers, features, and capabilities at the program level and issue type naming has been simplified to use epic for the purpose of this solution. An organization may choose to follow any other naming conventions suitable to their needs.
- Overall Jira Software configuration has been simplified for the purpose of this solution. Each Jira Software project can be further configured to have its own workflow, field, screen, permission, and notification schemes to align with enterprise needs.
- Overall Confluence configuration has been simplified for the purpose of this solution elicitation. Confluence can further be configured to have tailored templates to align with an organization's preference and documentation standards for a library of solution intent. SAFe® recommends this library to include present and future state representation of functional specifications, non-functional requirements, design documents, and test cases.

Team level

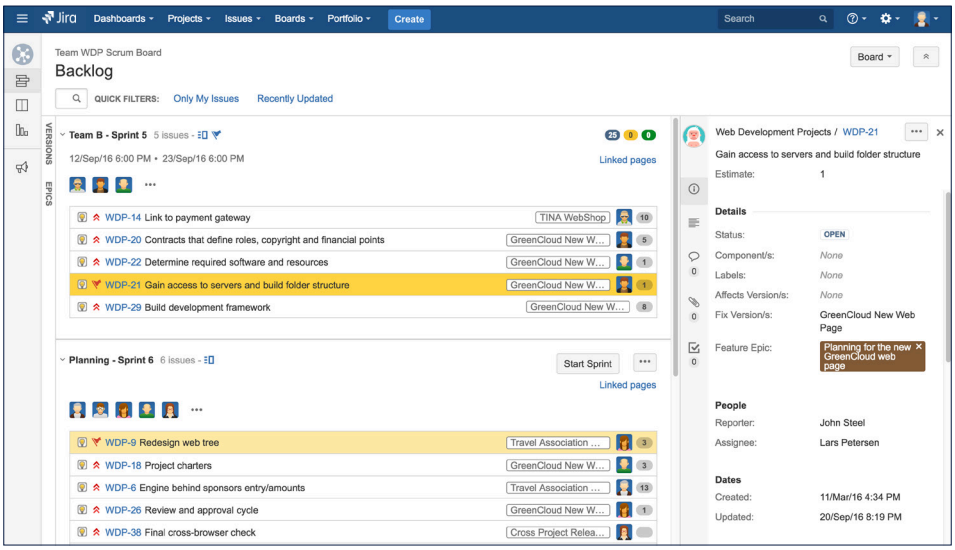
At the team level, the main goal is to prioritize work, assign work across teams based on needed skills and capacity, coordinate activities/manage dependencies, and perform what-if analysis for optimal throughput using available scope, resources, and time. Here are the key activities performed at the team level:

SCRUM MASTERS

Activities	Scrum Masters to create team specific agile boards
Tools used	Jira Software scrum board per team
Configuration	• Create a Jira Software project per team • Create a corresponding Jira Software scrum board for each of these project spaces

In our solution we have 3 teams:

- Team Wikk scrum board
- Team WDP scrum board
- Team cloud scrum board



Project scrum board backlog view in Jira Software

PRODUCT OWNER

Activities

Product Owner will maintain a team backlog based on the output of the PI planning session. This backlog may contain new stories, defects, and refactoring, design, and technology updates.
The following activities may be performed by the Product Owner:

- Break down user stories further into smaller deliverables
- Prioritize backlog items
- Refine acceptance criteria and size backlog in weekly backlog refinement meetings

Tools used

Jira Software scrum board per team

Configuration

In our solution we have three team scrum boards to plan, manage and deliver their sprint work:

- Team Wikk scrum board
- Team WDP scrum board
- Team cloud scrum board

PRODUCT OWNER, SCRUM MASTER, AND THE AGILE TEAM

Activities

Product Owner, Scrum Master, and the Agile Team will hold iteration planning meeting at the beginning of each sprint. Following activities are performed:

- Scrum Master and team will establish the available capacity or historic velocity for the sprint. This will serve as the objective anchor for the team's commitments.
- Product Owner will review the higher priority items in the backlog. Agile Team discusses solution options, technical constraints, non-functional requirements, and dependencies. This activity results in a more elaborated acceptance criteria and refined story points—both of which are captured at the story level in Jira Software.
- Agile Team will take these stories and further break it down into sub-tasks with assignees and original estimates—all of which are

Tools used

Jira Software scrum board per team - backlog

Configuration

In our solution, we propose the following configuration:

- Stories are sized using story points
- Tasks are broken down as subtasks of the stories
- Each subtask requires an assignee and original estimates
- Each subtask workflow requires time spent upon resolution

AGILE TEAM ITERATION EXECUTION

Activities

Once the Agile Team moves to iteration execution, they work to deliver their committed goals. Following activities are performed:

- **Agile Team** members use the active sprint area of their scrum board to manage their assigned work, flag issues, and report progress.
- **Scrum Master** uses the burndown chart in the reports area of their scrum board to track sprint health and progress of the sprint

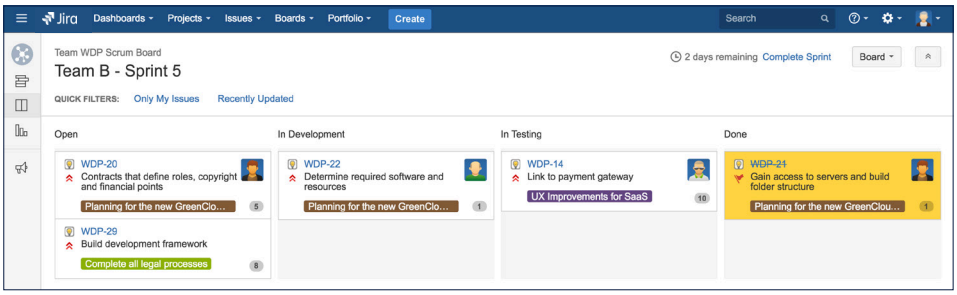
Tools used

Jira Software scrum board - active sprint and report views

Configuration

In our solution, we propose the following configuration:

- Agile Boards workflow is configured with an easy open, in development, testing, done workflow
- Agile Boards have quick filters configured for “Assigned To Me”
- Jira Software reports are available for each scrum board and include burndown charts and other reports



Project scrum board active sprint in Jira Software

Project scrum board burndown view in Jira Software

SCRUM MASTERS, PRODUCT OWNERS, AND AGILE TEAM

Activities

Scrum Masters, Product Owners, and Agile Team engages in continuous discussion and communication using instant messaging solutions for real-time updates and resolution on open issues. Additionally, daily stand-ups can be leveraged to review progress and burndown as well as to identify bottlenecks. As teams delivered each of their respective solutions, they can perform an integrated demo, also known as a system demo. stories meet iteration level DoD and moved to the appropriate status if system demo was successful.

Tools used

Hipchat

Configuration

Hipchat room for each team and cross team PI planning communication and collaboration

SCRUM MASTER

Activities

Scrum Master closes sprint once the iteration ends. Following activities can be performed:

- Sprint is ended and remaining stories moved to the backlog
- Sprint retrospective notes are captured and linked to the sprint

- Sprint report is reviewed in detail to understand the overall burn-down, scope changes, stories that were not completed as planned
- Team velocity chart is reviewed to understand team's average run rate and throughput trends

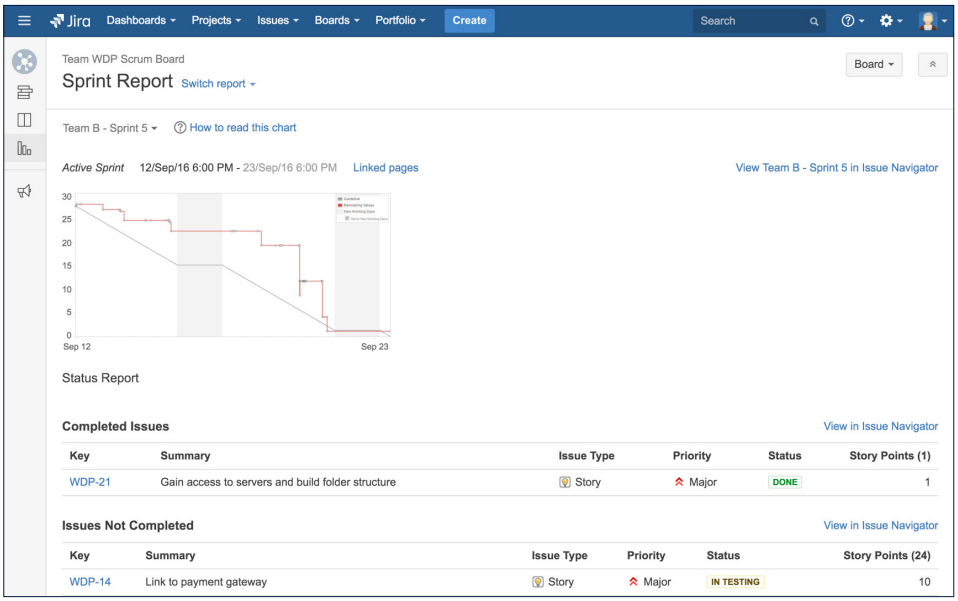
Tools used

- Jira Software scrum board
- Retrospective blueprint in Confluence

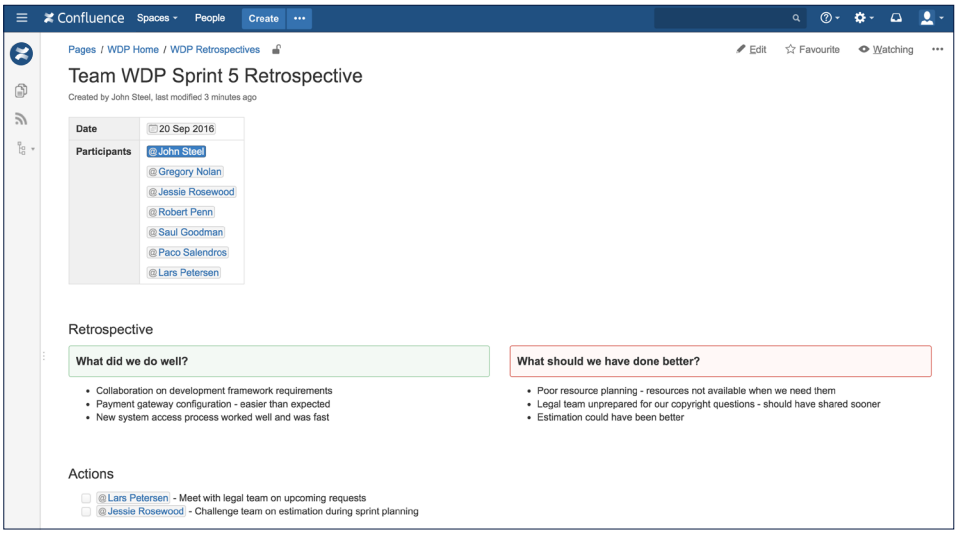
Configuration

No additional configuration required

- Use existing scrum board and Confluence configuration



Scrum board sprint report in Jira Software



Retrospective blueprint in Confluence

Team Level Assumptions:

- It is assumed that a requirement at the team level has already been through the PI planning and prioritization process so it is available to a specific team backlog.
- The solution highlights how Atlassian can be used as a solution for multiple teams in one PI and one ART. The same logic can be repeated for additional/multiple ARTs.
- The solution assumes that all teams are operating as scrum teams. You may have scrum and kanban teams as part of your ART and this solution can leverage Jira Software boards for each need. Portfolio for Jira will allow you to have mixed methodology teams.
- SAFe® recommends program epics, program enablers, features, and capabilities at the program level and issue type naming has been simplified to use epic for the purpose of this solution. An organization may choose to follow any other naming conventions suitable to their needs.
- Overall Jira Software configuration has been simplified for the purpose of this solution. Each Jira Software project can be further configured to have its own workflow, field, screen, permission and notification schemes to align with enterprise needs.
- Overall Confluence configuration has been simplified for the purpose of this solution elicitation. Confluence can further be configured to have tailored templates to align with an organization's preference and documentation standards for a library of solution intent. SAFe® recommends this library to include present and future state representation of functional specifications, non-functional requirements, design documents, and test cases.

Closing

This solution provides an approach to apply SAFe® 4.5 concepts and principles using the Atlassian suite. While this approach isn't prescriptive, it offers a way for teams to manage the SAFe® activities at each level and leverage the flexibility of Jira Software. This solution provides portfolio teams a way to plan and analyze work, program teams to break down work and allocate resources, and for teams to use an agile approach to delivering their work.

APPENDIX

[1] http://www.scaledagileframework.com/about/

[2] http://www.scaledagileframework.com/portfolio-level/

[3] http://www.scaledagileframework.com/large-solution-level/

[4] http://www.scaledagileframework.com/program-level/

[5] http://www.scaledagileframework.com/team-level/

[6] http://www.scaledagileframework.com/agile-release-train/

[7] http://www.scaledagileframework.com/program-increment/

[8] http://scaledagileframework.com/WSJF

[9] http://scaledagileframework.com/WSJF

[10] http://scaledagileframework.com/lean-agile-leaders

[11] http://scaledagileframework.com/communities-of-practice/

[12] http://scaledagileframework.com/program-portfolio-management/

[13] http://scaledagileframework.com/lean-portfolio-management/

[14] http://scaledagileframework.com/epic-owner/

[15] http://scaledagileframework.com/enterprise-architect/

[16] http://scaledagileframework.com/release-train-engineer-and-value-stream-engineer/

[17] http://scaledagileframework.com/product-and-solution-management/

[18] http://scaledagileframework.com/system-and-solution-architect-engineering/

[19] http://scaledagileframework.com/customer/

[20] http://scaledagileframework.com/release-train-engineer-and-value-stream-engineer/

[21] http://scaledagileframework.com/system-and-solution-architect-engineering/

[22] http://scaledagileframework.com/product-and-solution-management/

[23] http://scaledagileframework.com/business-owners/

[24] http://scaledagileframework.com/product-owner/

[25] http://scaledagileframework.com/scrum-master/

[26] http://scaledagileframework.com/agile-teams/

[27] http://scaledagileframework.com/strategic-themes/

[28] http://www.scaledagileframework.com/value-streams/

[29] http://scaledagileframework.com/agile-release-train/

[30] http://scaledagileframework.com/pi-planning/

[31] http://scaledagileframework.com/program-increment/

[32] http://scaledagileframework.com/solution-demo

[33] http://scaledagileframework.com/iteration-planning/

[34] http://scaledagileframework.com/iteration-execution/

[35] http://scaledagileframework.com/system-demo/

[36] http://scaledagileframework.com/iteration-retrospective/

Team level agility

Scaling agile at the organizational level provides many of the same benefits to the overall organization as it does to individual teams who have embraced agile. These benefits include reduced time to market, nimbleness in the development process, and more transparency across the organization.

The two biggest challenges in scaling agile often occur at opposite ends of the spectrum. These challenges are letting go of waterfall defaults which:

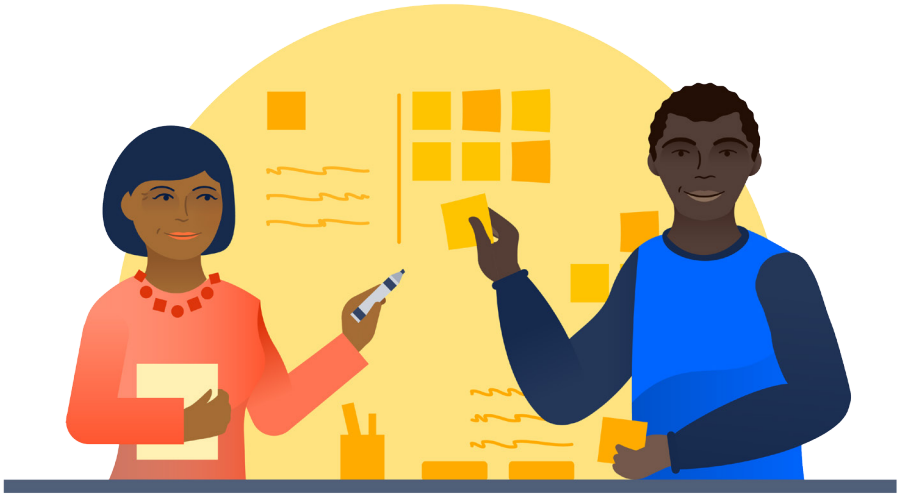
1. Encourage organizational rigidity
2. Don't embrace the culture of agility throughout the development process

Let's take a look at the practices that guard against these challenges and promote a healthy basis for scaling agile.

Team agile 101

Many teams often fall into the trap believing they are agile because they do stand-ups. Unfortunately, agile is not waterfall development plus stand-ups. While stand-ups are an important ceremony in agile, they are one of many changes that an organization goes through (albeit one of the most recognizable). Agile is an iterative methodology where new features and feedback are incorporated regularly. We see agile adoption as a cultural change, not a process. Agile is first and foremost a cultural change, which will have an impact on your processes and tools. Teams must have the trust of the organization and the courage to make changes to their process as they learn more about the product, development realities and customers they serve.

> "We see agile adoption as a cultural change, not a process.



How do these cultural items manifest themselves? Let's focus on scrum, as it's the most popular agile framework. We'll use some of the reports in Jira Software to highlight these best practices.

Team ceremonies

First, look to see that the individual teams are doing the basic scrum ceremonies: sprint planning, sprint reviews, sprint retrospectives, and stand-ups. Secondly, dig deeper. Following process in isolation without learning and growth doesn't help the team to be agile. Here's where to look to see if the team is growing:

- Ask the scrum master how feedback from the team's last few retrospectives shaped the following sprint planning meetings
- Ask the product owner what they learned about the product in the last sprint review
- Ask the team how the sprint review and retrospective affected the team's backlog
- Ask some of the engineers on the team how stand-up helps them ensure the team meets its goals

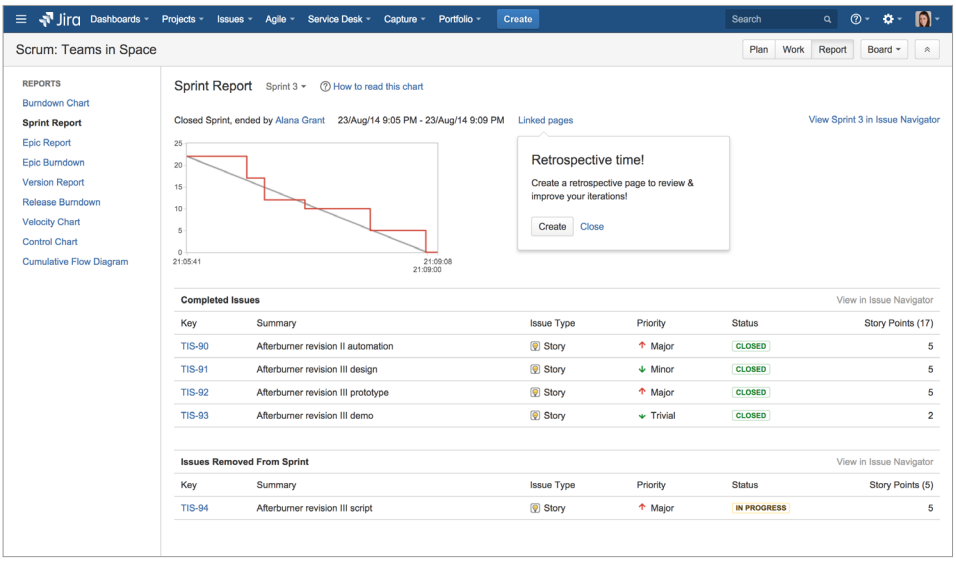
Doing a ceremony is not enough. The team needs to see value in each ceremony and each ceremony needs to affect the team's approach and direction on a regular basis.

Team metrics

The next step in assessing a team's agile health is looking at the metrics they find valuable that shape their development process. The team should use several reports inside of Jira Software to understand the quantitative side of development. Look to the following reports for each team inside of the portfolio:

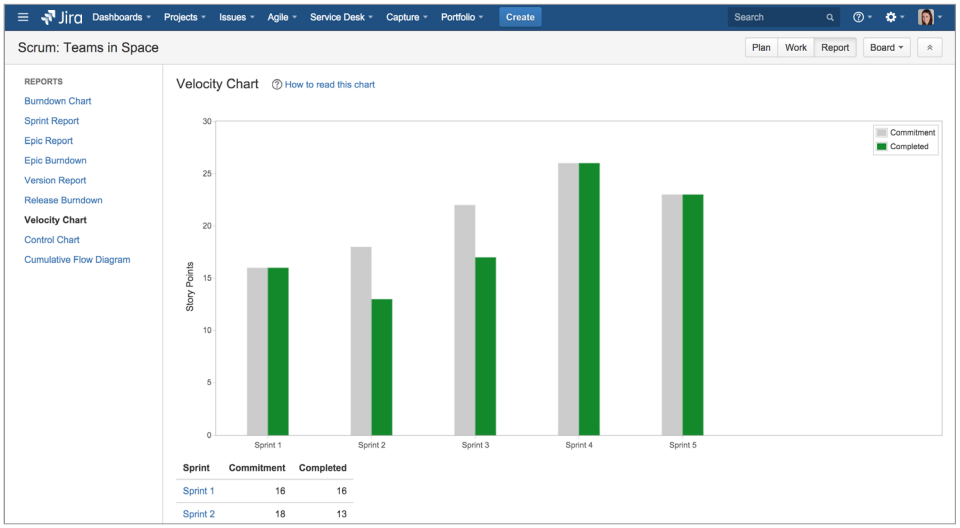
Sprint report:

- Did the team deliver on their forecast?
- Did the team honor the integrity of the last few sprints? How many unanticipated scope changes crept in?
- Does the team have a natural burndown or does everything crash to zero at the end of the sprint?



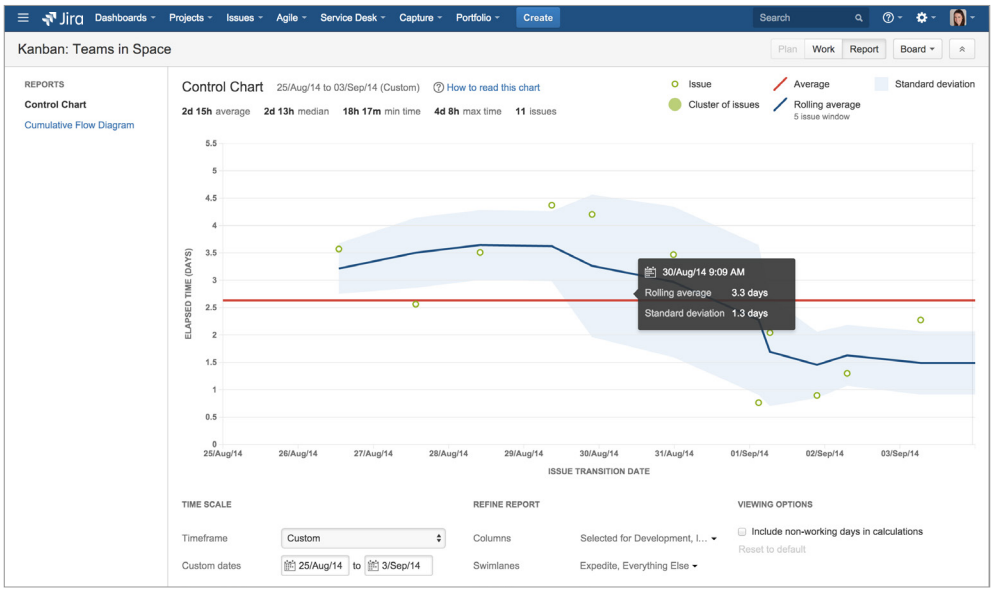
Velocity Chart:

- Does the team have a consistent velocity over the last several sprints? If not, is it clear why and does the team have a path forward?
- Is the team regularly delivering what they forecasted over the last few sprints?



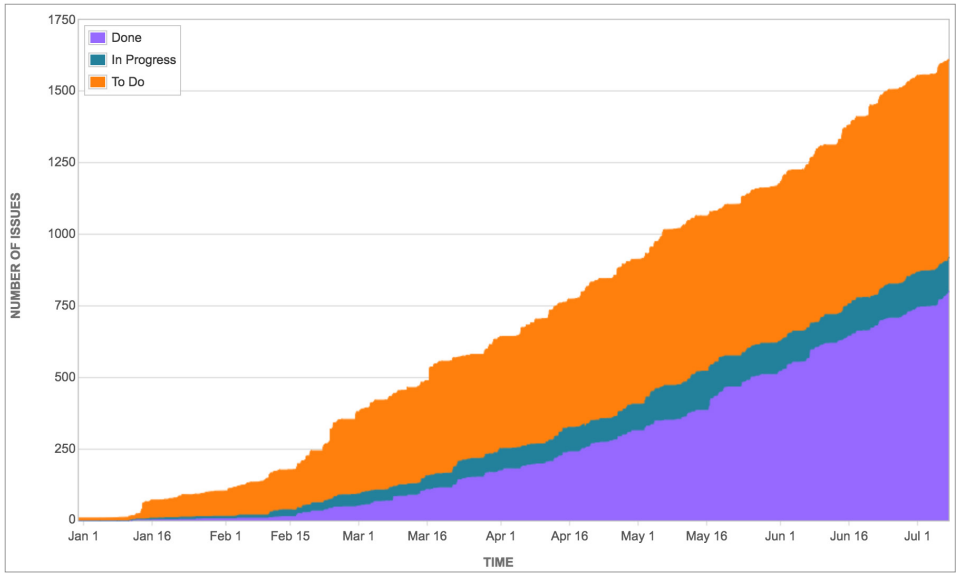
Control Chart:

- Over the last three months of development is the cycle time of each story point value consistent? For example, does the story point value of 5 have a consistent cycle time?
- Are there any long-running stories that are not getting resolved?



Cumulative flow diagram:

- Does work flow evenly across the team?
- Are there any states where work backs up and starves a future state?



Don't single out a team on a particular metric. The question to ask is whether the team uses each of these charts to inform and evolve its development process. If so, they are embracing agility and growing together over time. Thus any localized abnormalities will resolve over time.

Organizational Culture

While the processes and metrics around agile are important, the culture of the individual teams in the organization reigns supreme. Culture is the toughest, most important and unique muscle to develop. Every geography, every department, every product team will be slightly different and that's ok. The key culture questions are:

- Do team members have genuine relationships with one another?
- Do team members share the workload to get to the common goal of an iteration?

- Does the team self organize around the work and deliver selflessly without politics?
- Does the team have a good working relationship (and appropriate boundaries) with the product owner and stakeholders?

Scaling agile also means scaling culture across the business. Teams are living entities too. Much like individuals need investment, so do teams. Questions to ask include:

- Does each team in the organization have a relationship with one another?
- Do team members know how to reach beyond team boundaries to get work done?
- Are skill sets and knowledge of the code base spread across the organization?
- Are estimation tactics aligned across the teams, and if you are using story points, are teams weighing a story point in a balanced fashion?

Unfortunately, there is no clear quantitative or qualitative methodology to ensure health before beginning a scaled agile effort. Much like agile itself, ensuring ongoing health of individual teams within the portfolio is an iterative endeavor. This is where retrospectives have a value that goes beyond the product development lifecycle. Many times, the most important cultural changes are introduced at the team level, and spread across to the rest of the organization. The same way that customers can inform Product Managers of the next best feature (which takes place in customer interviews), your team can inform the organization of the how to maintain health at the team level (and this takes place during retrospectives).

ATLASSIAN + CPRIME