

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





SCALING AGILE WITH ATLASSIAN AND SAFE

CONTENTS

Introduction and objectives	3
The need to scale agile	4
Team level agility	6
SAFe® overview	12
Tool requirements to support SAFe®	18
Atlassian and SAFe®	19
Solution overview	20



SCALING AGILE WITH ATLASSIAN AND SAFE

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 and organizational needs at all levels.

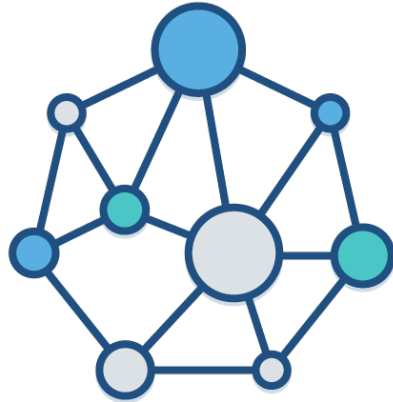


The need to scale agile

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 which combine multiple features across different teams from concept to delivery
- Plan and align business value 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®) 1.0 to help address these issues and provide success patterns for large and small organizations.

SAFe® is now in its 4th iteration (SAFe® 4.0) and has been updated based on customer feedback, and real world usage patterns. It also enjoys the widest adoption as a method to scale agile.

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

to prioritize, estimate, and deliver new features to customers and then incorporate feedback 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?

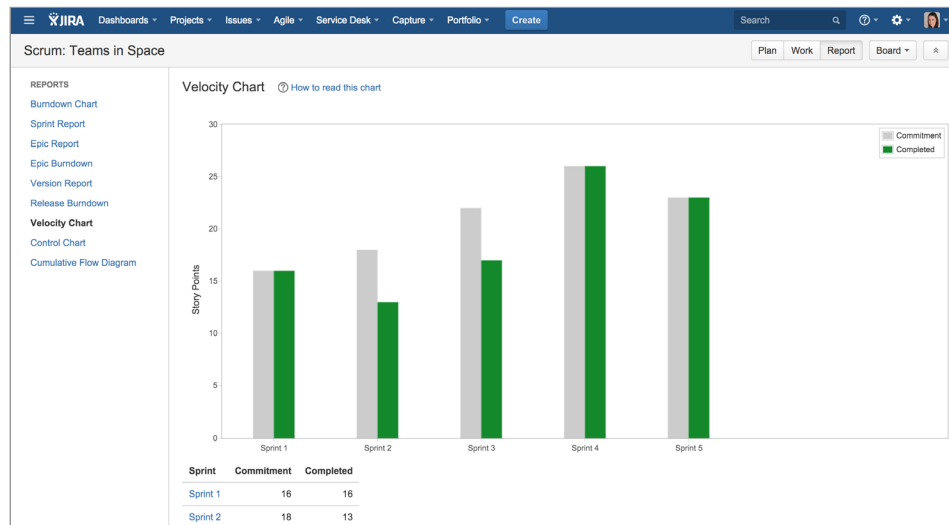
The screenshot shows the JIRA Sprint Report interface. At the top, there's a navigation bar with 'JIRA' and various menu items. Below that, the page title is 'Scrum: Teams in Space'. The main content area is titled 'Sprint Report' for 'Sprint 3'. It features a burndown chart showing progress over time, a 'Retrospective time!' notification box with 'Create' and 'Close' buttons, and two tables: 'Completed Issues' and 'Issues Removed From Sprint'. The 'Completed Issues' table lists four items with their keys, summaries, issue types, priorities, statuses, and story points. The 'Issues Removed From Sprint' table lists one item with its key, summary, issue type, priority, status, and story points.

Key	Summary	Issue Type	Priority	Status	Story Points (17)
TIS-90	Afterburner revision II automation	Story	Major	CLOSED	5
TIS-91	Afterburner revision III design	Story	Minor	CLOSED	5
TIS-92	Afterburner revision III prototype	Story	Major	CLOSED	5
TIS-93	Afterburner revision III demo	Story	Trivial	CLOSED	2

Key	Summary	Issue Type	Priority	Status	Story Points (6)
TIS-94	Afterburner revision III script	Story	Major	IN PROGRESS	5

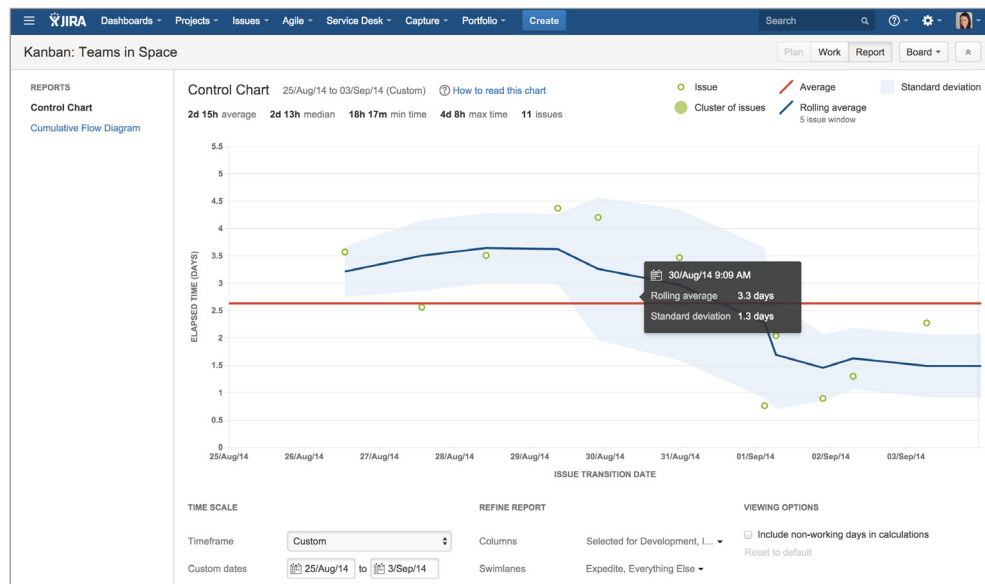
Velocity Chart:

- 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?



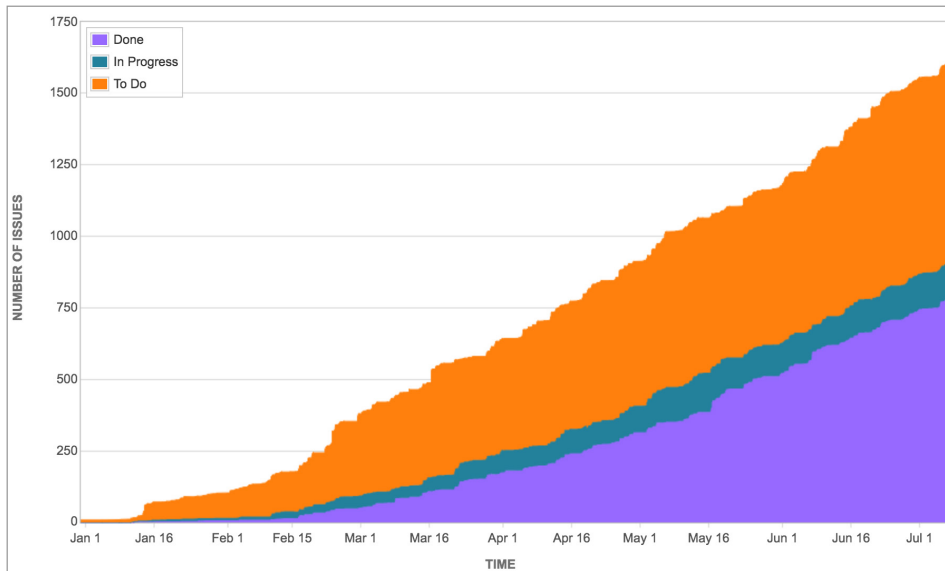
Control Chart:

- 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).



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.

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. This whitepaper closely aligns to the SAFe® 4.0 diagram found at <http://www.scaledagileframework.com/>. For questions on terminology, please refer to that document.

- **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
- **Program level** ³ – where team resources are applied to critical, ongoing development work
- **Team level** ⁴ – 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 **two core constructs**:

- **Agile Release Train (“ART”)** ⁵ - 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”)** ⁶ - a timebox that synchronizes planning, delivery, and reviews within an ART

SAFe® recommends WSJF ⁷ as a way to objectively evaluate, weigh, and prioritize your epics in the backlog. Weighted Shortest Job First (WSJF) = CoD / Job Size.

- $\text{Cost of Delay}(\text{CoD}) = \text{Business Value}(\text{BV}) + \text{Risk Reduction}(\text{RR}) + \text{Time Criticality}(\text{TC})/\text{Opportunity Enablement}(\text{OE})$
- 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 ⁷).



While we expect to have lean-agile leaders⁸ and a community of practice⁹ driving SAFe® adoption and usage, these are some of the most important roles at each level:

Portfolio level

Program portfolio management¹⁰ – responsible for strategy, investment, governance, and program management. Primary roles are:

- Epic owner – responsible for identification and creation of epics aligned by vision and strategic themes. Create value statements and lightweight creative business case. Owns portfolio backlog, WSJF prioritization, and go/no-go decisions for completed work¹¹.
- Enterprise architect¹² – responsible for ensuring enterprise-wide architectural system, platform, and infrastructure dependent work is identified and created

Value stream level (optional)

- Value stream engineer¹³ – responsible for facilitating value stream and ART processes and execution
- Solution management¹⁴ – responsible for value stream backlog content
- Solution architect/engineer¹⁵ – responsible for alignment with enterprise architecture, as well as identifying and creating solution architecture
- Customer¹⁶ – internal or external resource responsible for planning, evaluating solution increments, review status of work, and provide testing/UAT feedback

Program level

- Release train engineer¹⁷ – responsible for facilitating value stream and ART processes and execution including PI Planning, alignment with vision, and value stream objectives

- System architect/engineer¹⁸ – responsible for alignment with enterprise and solution architecture, and identifying and creation of solution architecture to be delivered by teams architecture
- Product management¹⁹ – responsible for product backlog content
- Business owner – responsible for fiduciary, governance, efficacy, and return on investment for an agile release train

Team level

- Product owner²⁰ – responsible for defining stories, prioritizing the team backlog, review, and accepting stories
- Scrum master²¹ – responsible for lean-agile leadership, agile process facilitation, enabling the team, and removal of impediments
- Scrum team²² – group of individuals responsible for defining, building, and testing components/features within their agile process

Each level will have specific ceremonies or activities to ensure appropriate inputs and outputs, that the process is sustainable, and to build in quality. These include:

Portfolio and value stream (optional) level

- Create strategic themes²³ – identify, analyze, and create strategic business goals and objectives
- Define value streams²⁴ – identify and create the value streams for your organization (if applicable)
- Define portfolio backlog - create portfolio epics aligned to vision and themes and objectively prioritized using WSJF



Program level

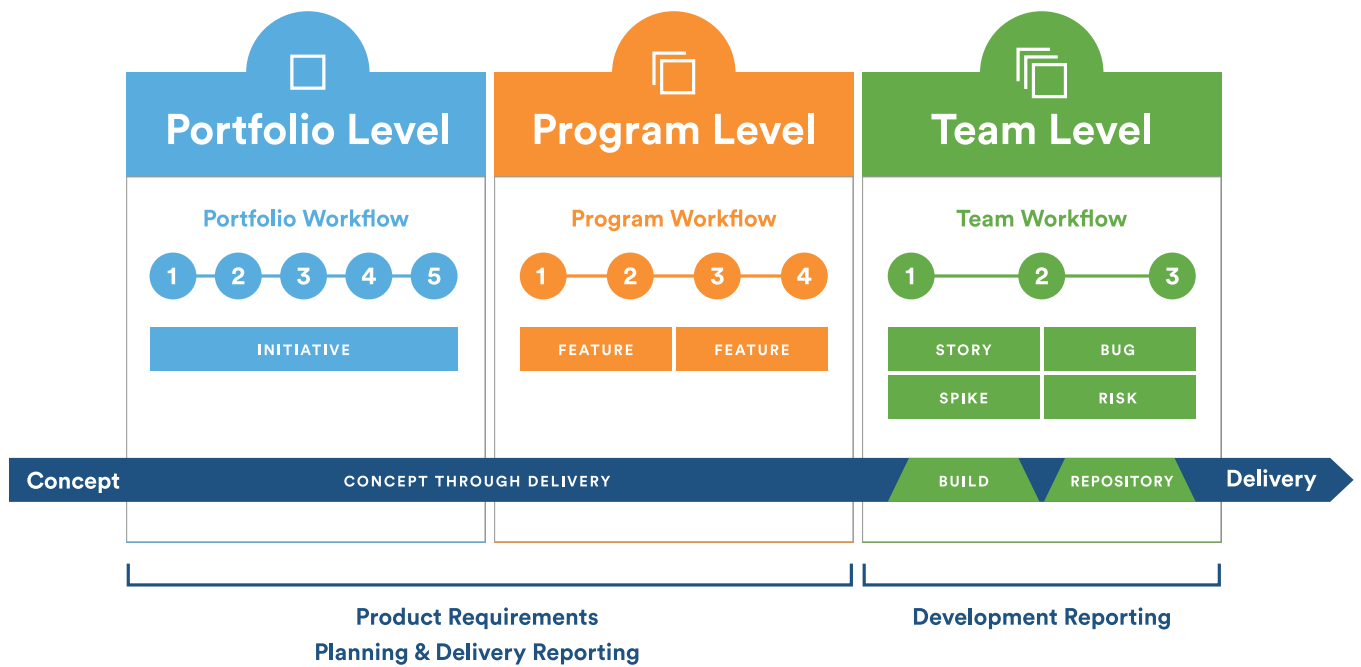
- Form and launch ARTs²⁵ – organize what teams, skillsets, product/service/feature a release train will deliver
- PI planning²⁶ – at the beginning of each PI, teams will plan the next PI by estimating work and identifying dependencies
- ART sync (scrum of scrums and PO sync)²⁷ – alignment meetings to ensure teams and POs are aligned on work progress, challenges, and next steps
- Solution demo²⁸ – at the end of each PI, teams will demonstrate the progress on the solution to key stakeholders

Team level

- Iteration planning²⁹ – before each iteration, teams and POs will organize their work and define a realistic scope for the upcoming iteration
- Iteration execution³⁰ – delivery, impediment resolution, and tracking of iteration progress
- System demo³¹ – at the end of each iteration, teams will demonstrate the progress on the solution to key stakeholders
- Iteration retrospective³² – a candid team review of what worked, what didn't, and what should be done differently each iteration

We'll focus on the roles and ceremonies in future sections as we show how to use the Atlassian suite to support the SAFe® methodology.

Scaling from concept through delivery





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 team collaboration software – including JIRA Software, Confluence, Bitbucket, HipChat, 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 so many different teams, reflecting the continued commitment to R&D.

Atlassian's JIRA Software, Portfolio for JIRA, Confluence, and HipChat 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



Solution 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.

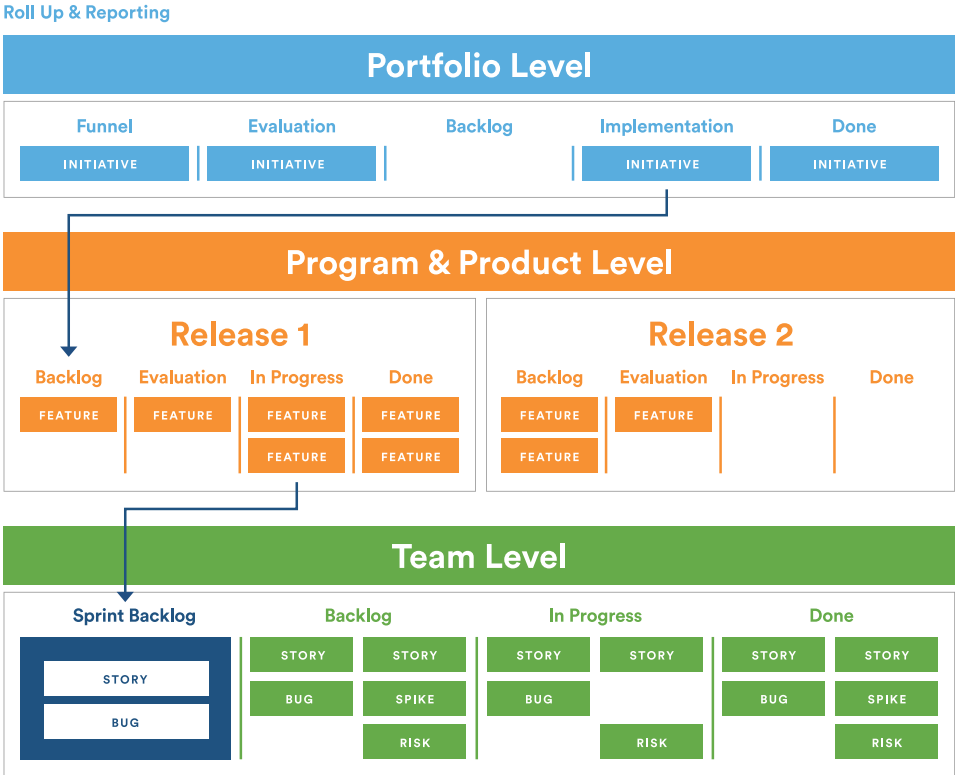
The high level tasks completed by each level in the organization include:

- **Portfolio level** - The Portfolio level focuses on idea generation which is captured through an intake funnel. These ideas are vetted for economic and business viability; the approved initiatives are evaluated at the program level.
- **Program and product level** - Your program teams break those initiatives into features, establish dependencies, estimate, and bundle 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 initiatives

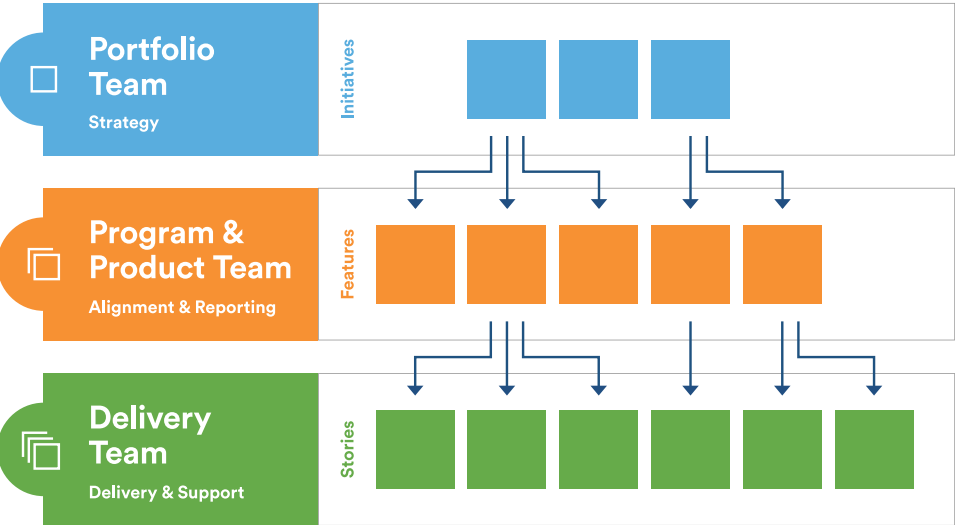
Let's take a closer look at the primary needs of each level and how they can accomplish these goals using the proposed tools.

Note: Issue type nomenclature has been simplified for the purpose of this solution. An organization may choose to follow any other issue type naming convention suitable to your culture and preference.

Scaling agile Atlassian solution - organization



Scaling agile Atlassian solution - hierarchy

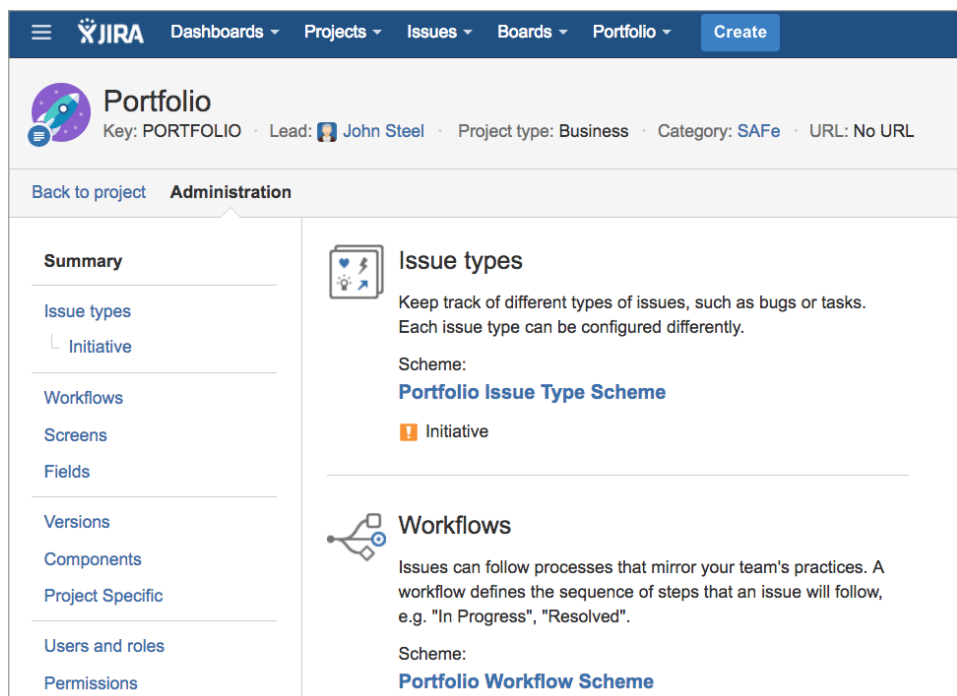


Portfolio level

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:

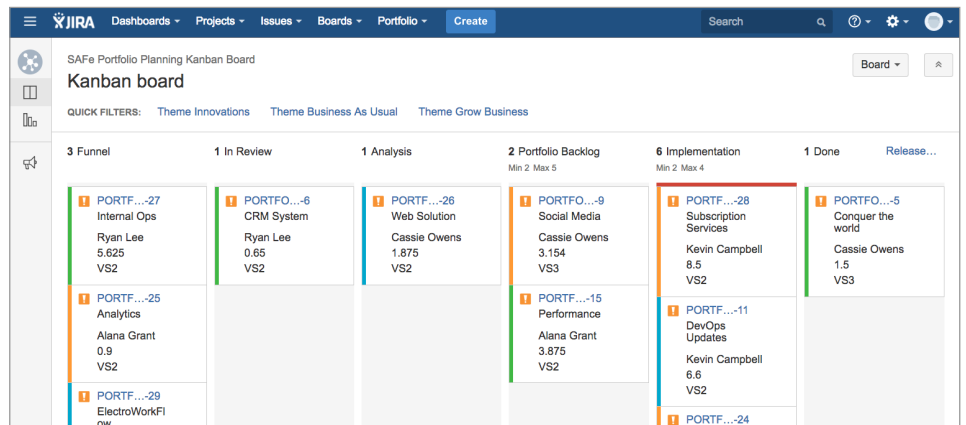
PORTFOLIO EPIC OWNERS

Activities	Portfolio Epic Owners submit their requests to the funnel. Additionally, Program Epic Owners can also create requests which can be associated with portfolio level epics.
Tools used	JIRA Software kanban board for intake and flow management
Configuration	<ul style="list-style-type: none">• One portfolio level JIRA project with an Initiative issue type• Portfolio level JIRA Software kanban board based on portfolio level project• One program level JIRA Software project with feature issue type• Program level JIRA Software kanban board based on program level project



The screenshot shows the JIRA Portfolio Administration interface. The top navigation bar includes 'Dashboards', 'Projects', 'Issues', 'Boards', 'Portfolio', and a 'Create' button. The main header displays 'Portfolio' with a key of 'PORTFOLIO', lead 'John Steel', project type 'Business', category 'SAFe', and URL 'No URL'. Below this, there is a 'Back to project' link and an 'Administration' tab. The left sidebar contains a 'Summary' section with links for 'Issue types', 'Workflows', 'Screens', 'Fields', 'Versions', 'Components', 'Project Specific', 'Users and roles', and 'Permissions'. The main content area is divided into two sections: 'Issue types' and 'Workflows'. The 'Issue types' section includes a description: 'Keep track of different types of issues, such as bugs or tasks. Each issue type can be configured differently.' It shows the 'Scheme: Portfolio Issue Type Scheme' and a single 'Initiative' issue type. The 'Workflows' section includes a description: 'Issues can follow processes that mirror your team's practices. A workflow defines the sequence of steps that an issue will follow, e.g. "In Progress", "Resolved".' It shows the 'Scheme: Portfolio Workflow Scheme'.

Portfolio kanban board in JIRA Software



EPIC OWNERS

Activities

Epic owners submit value statements and lightweight business case

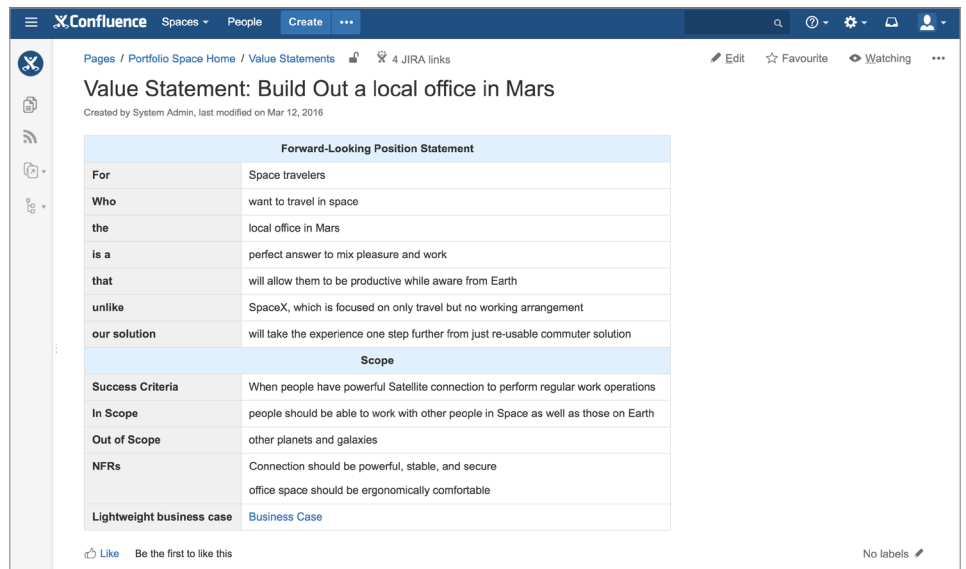
Tools used

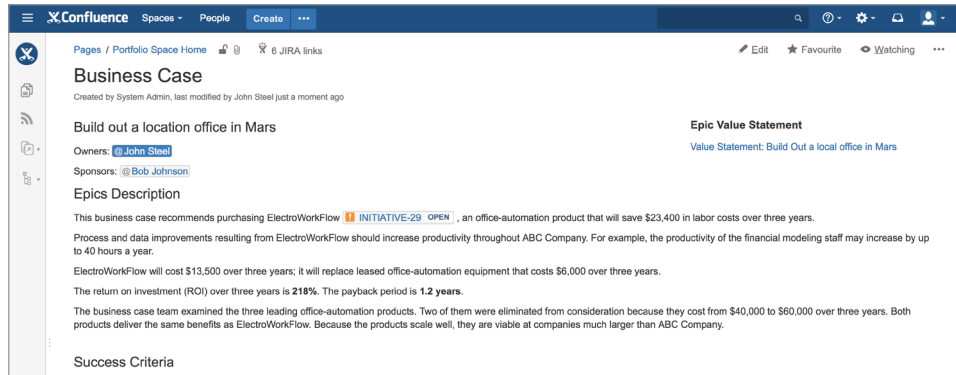
Confluence for template based documentation

Configuration

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

Confluence value statement





Confluence business case

SOLUTION MANAGEMENT TEAM (EVALUATION)

Activities

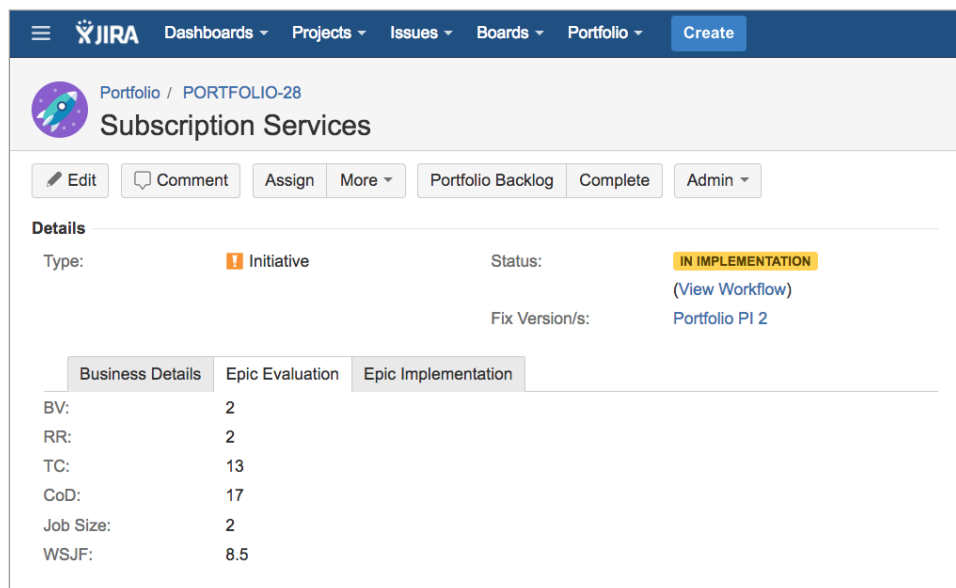
Solution Management Team evaluates the alignment of the epic with strategic themes and vision, approves or rejects the initiative, and finally prioritizes it for execution using relative WSJF measure

Tools used

JIRA Software custom calculated fields for WSJF calculation

Configuration

- Configure JIRA Software custom fields for intake measures such as BV, RR, TC, CoD, Job Size, and WSJF
- Configure WSJF as JIRA Software misc calculated field to compute the score based on the input of the other custom fields



Calculated WSJF issue in JIRA Software

SOLUTION MANAGEMENT TEAM (EXECUTION)

Activities

Once prioritized and moved to execution, Solution Management Team will define capabilities and decompose features. Activities include:

- Create hierarchy of initiatives and features
- Breakdown initiatives into features
- Map PI cross-project release (CPR) to the features
- Sync information with JIRA Software (automatically or live)

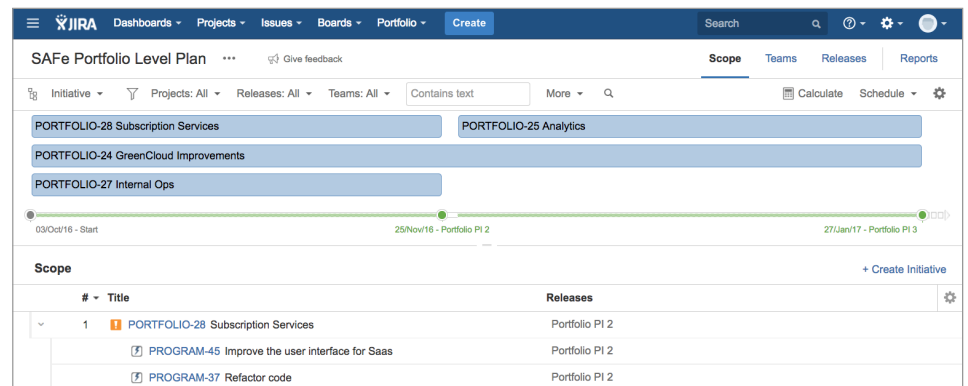
Tools used

Portfolio for JIRA for feature breakdown

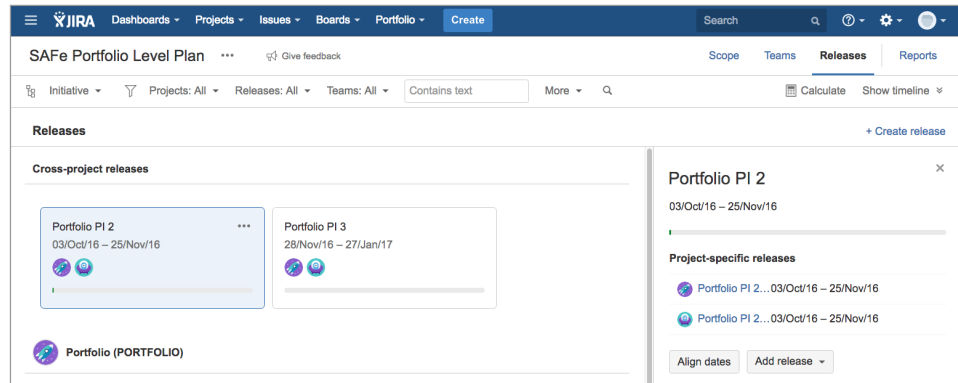
Configuration

- One portfolio level plan with portfolio and program projects (no boards or filters)
- Need a cross product release per program increment (PI) to allow portfolio and program teams to assign issues to an PI
- NOTE: Schedule and team capacity functionality will be leveraged at the program level

Portfolio plan overview in Portfolio for JIRA



*Portfolio plan CPR in
Portfolio for JIRA*



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 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 portfolio epics and portfolio enablers and issue type naming has been simplified to use initiatives 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. Confluence can further be configured to have tailored templates to align with organizational needs and documentation standards.

Program level

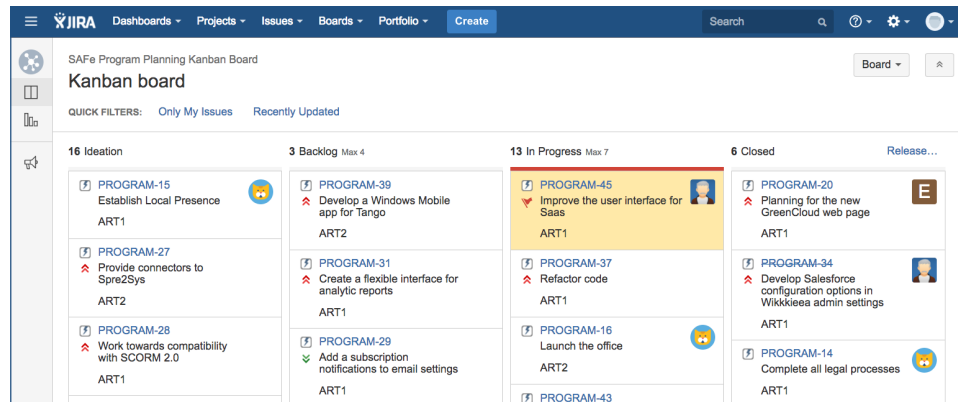
At the program level, the main goal is to prioritize work, assign work across teams based on needed skills and capacity, coordinate activities, manage dependencies, and perform what-if analysis for optimal throughput using available scope, resources, and time. Here are the key activities performed at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

PROGRAM AND PORTFOLIO MANAGEMENT TEAM

Activities	Program and portfolio management team manage demand and continuous flow of value using the program board
Tools used	<ul style="list-style-type: none"> • Portfolio for JIRA for feature breakdown • JIRA Software
Configuration	<ul style="list-style-type: none"> • Create a JIRA Software project space for program level epics • Create a corresponding JIRA Software kanban board for this to visually track and manage the flow of your epics

The screenshot displays the JIRA Program Administration interface. At the top, the navigation bar includes 'Dashboards', 'Projects', 'Issues', 'Boards', 'Portfolio', and a 'Create' button. The main header shows the 'Program' name, key (PROGRAM), lead (Erica Jefferson), project type (Business), category (SAFe), and URL (No URL). Below this, the 'Administration' section is active, showing a sidebar with various configuration options. The main content area is divided into two sections: 'Issue types' and 'Workflows'. The 'Issue types' section explains that it's used to track different types of issues like bugs or tasks, and lists the 'Program Issues Type Scheme' with two types: 'Epic' and 'Initiative'. The 'Workflows' section explains that it defines the sequence of steps an issue will follow, such as 'In Progress' or 'Resolved', and lists the 'Program Workflow Scheme' with one workflow: 'SAFe Program Workflow'.

Program level project in JIRA Software



Program kanban board

PRODUCT MANAGERS

Activities

Product Managers bring their respective backlog of approved initiatives and epics for PI planning

Tools used

Portfolio for JIRA for PI planning and story breakdown

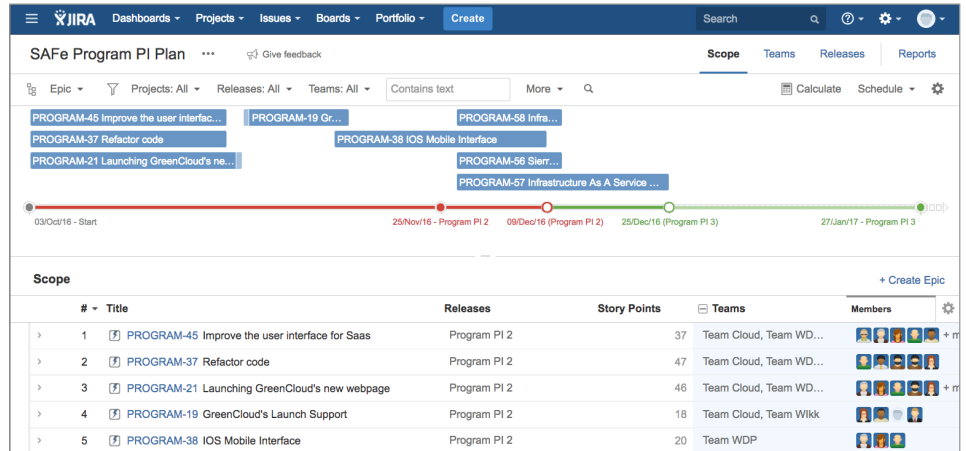
Configuration

Create program level plan based on:

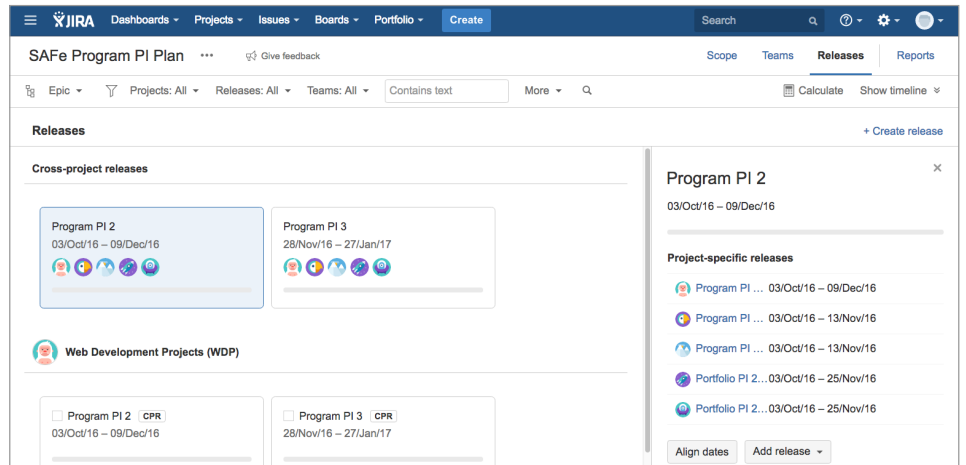
- All team boards (refer to team overview)
- A filter of items of all epics mapped to fix version for PI specific CPR (created in portfolio level portfolio plan)
- A filter of items of all initiatives mapped to fix version for PI specific CPR (created in portfolio level portfolio plan)
- Create shared or private scrum teams with relevant team members, default velocity, estimated capacity per team member and map to each of the respective scrum board
- Create 1 CPR for the specific PI and map to team project spaces only (this would include issues from various ARTs within a value stream) releases will have cross project releases tied to incremental releases during a PI and associate individual project fix versions with a CPR

- Create strategic themes and allocation
- Exclude initiatives without any work in a PI

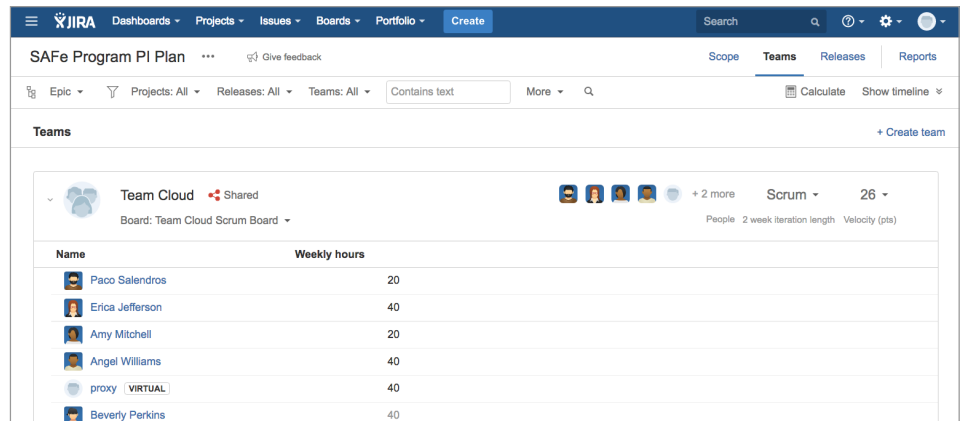
PI plan overview in Portfolio for JIRA



PI plan CPR/ART in Portfolio for JIRA

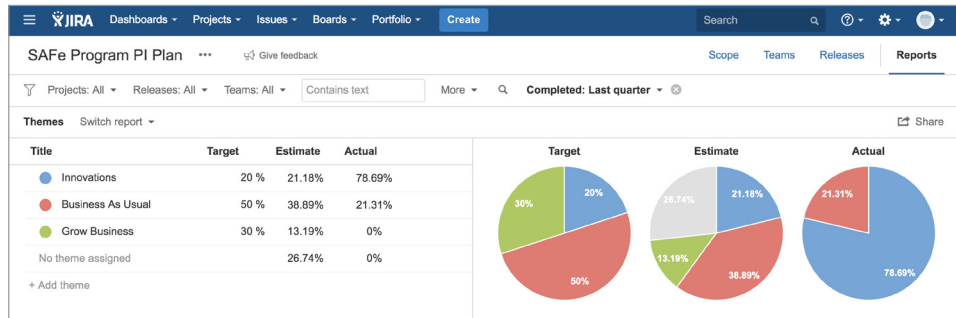


PI plan team view in Portfolio for JIRA





PI plan themes in Portfolio for JIRA



PRODUCT OWNER

Activities

Product Owner will break epic into stories, estimate story value, and prioritize backlog

- Break down stories for each prioritized epic
- Provide estimates in story points
- Map to strategic themes and team
- Map to CPR for the PI

Tools used

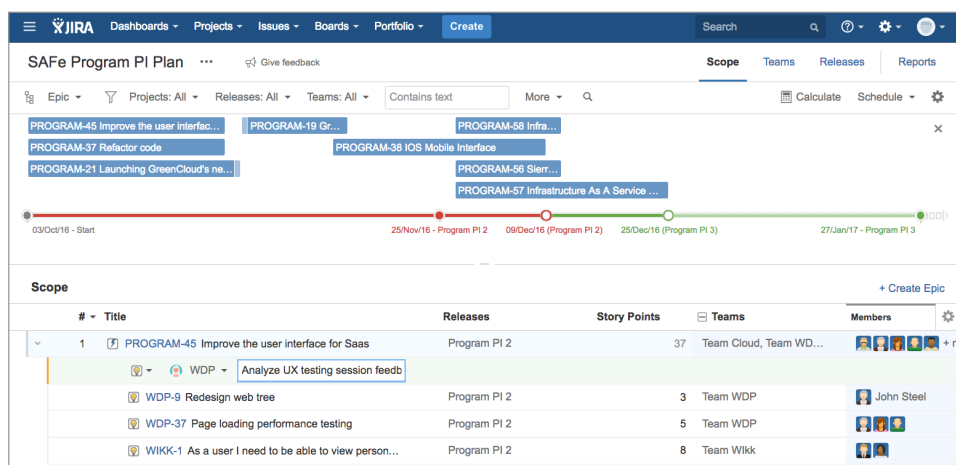
Portfolio for JIRA for PI planning and story breakdown

Configuration

No additional configuration required

- Use existing Portfolio for JIRA program level plan

Epic breakdown in Portfolio for JIRA



PI PLANNING TEAM

Activities

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

- For a list of epics mapped to cross-project release for the current PI, you may find the initial roadmap shows in red. This means that the PI is currently overbooked and won't ship on the expected release date.
- You have an option to add more capacity, or move the release date, or reduce scope. In our example, we moved two epics out of the scope of current PI. The revised roadmap now shows in green because scope, time, and capacity are all in perfect balance.

Tools used

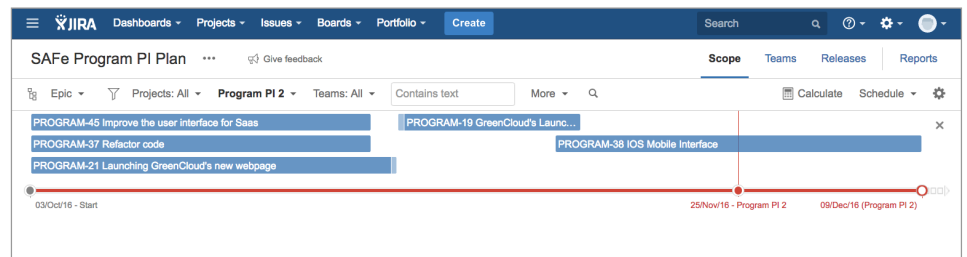
Portfolio for JIRA for PI planning and story breakdown

Configuration

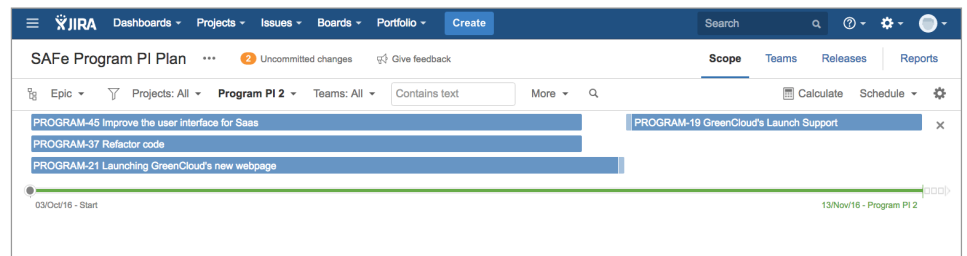
No additional configuration required

- Use existing Portfolio for JIRA Program level plan

PI schedule in Portfolio for JIRA (plan overbooked)

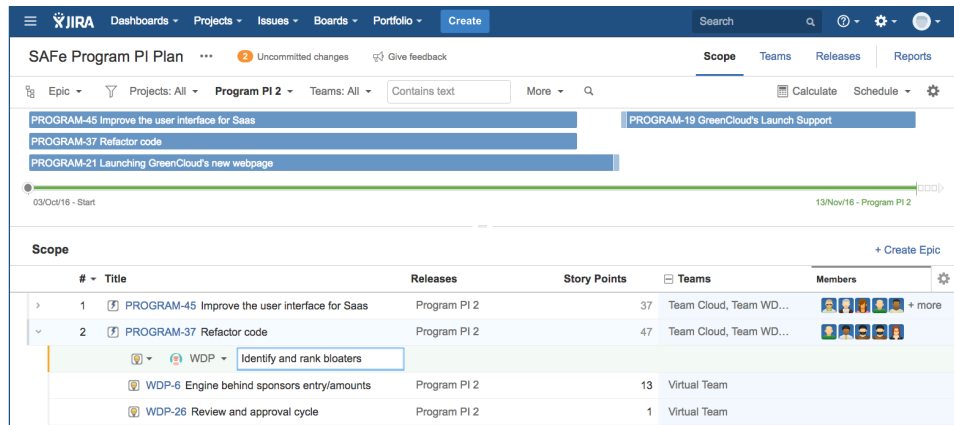


PI schedule in Portfolio for JIRA (plan on track)



RELEASE TRAIN ENGINEER

- Activities** Once prioritized, Release Train Engineer may move the plan to execution
 - Sync information with JIRA Software for the stories to show up in each of the team’s relevant boards (automatically or live)
- Tools used** Portfolio for JIRA to promote centralized plan over to the team for decentralized decision making
- Configuration** No additional configuration required
 - Use existing Portfolio for JIRA program level plan



Epic breakdown to story 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 space 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

The screenshot shows a Confluence page titled "PI 2 Risk Register" with a table containing three risk entries. The table has columns for ID, Risk Name, Impact, Likelihood, Associated Feature, Owner, and Analysis.

ID	Risk Name	Impact	Likelihood	Associated Feature	Owner	Analysis
1	Payment gateway not prepared for launch	MEDIUM	LIKELY	PROGRAM-21 - Launching GreenCloud's new webpage IN PROGRESS	@Erica Jefferson	Greencloud Launch Risks
2	UI Vendor unable to prepare wireframes	HIGH	UNLIKELY	PROGRAM-38 - IOS Mobile Interface IN PROGRESS	@David Carsen	DesignWorks resources assigned to other internal project, can only create half of the wireframes in time for team review
3	Profile management process has no internal owner	MEDIUM	UNSURE	PROGRAM-42 - Site profiles IN PROGRESS	@Summer Robinson	Various IT and UI teams are working on this but a lack of owner has made it hard to determine if this will be complete in time for roll out in PI 2



2016.09.20 - Wikk Team Meeting Notes

Created by John Steel, last modified just a moment ago

Agenda / Table of Contents

- Attendees
- Agenda
- Notes
- Development Process
- Cross Team Update
- Action Items

Attendees

- @Gregory Nolan
- @Jessie Rosewood
- @Robert Penn
- @Saul Goodman
- @Paco Salendros

Agenda

- Review last meetings action item updates
- Discuss Development Pipeline process
- Cross-team Updates
- Action item allocation

Notes

- WDP team needs update on Bitbucket licensing to account for more teams - @Paco Salendros to follow up on 24 Sep 2016
- Delay on IOS Icon may impact system demo this iteration - @Gregory Nolan to discuss impact with Program Mgr 23 Sep 2016
- Code refactoring going well, need to continue to dedicate time to it each day to stay on track. @Saul Goodman to help with any outside resource requests and escalations.

Page view of meeting notes in Confluence

PI 2 Objectives

Created by John Steel, last modified just a moment ago

Program Increment Details

Iterations	Iteration 9 - 12
Stakeholder	@Erica Jefferson
Agile Release Train	Exploration ART
Teams	View teams in Portfolio for JIRA <ul style="list-style-type: none"> Wikk Cloud WDP

Program Increment Objectives

General PI Objectives	Complete site upgrades and enable core subscription services
Wikk Team Objectives	<ul style="list-style-type: none"> Update site profiles [PROGRAM-42 - Site profiles IN PROGRESS] Update IOS mobile icons [PROGRAM-38 - IOS Mobile Interface IN PROGRESS] Refactor Code [PROGRAM-37 - Refactor code IN PROGRESS]
Cloud Team Objectives	<ul style="list-style-type: none"> Launch Greencloud landing page, product pages, and purchase pages [PROGRAM-21 - Launching GreenCloud's new webpage IN PROGRESS] Collect UAT on beta versions
WDP Team Objectives	<ul style="list-style-type: none"> Improve SAAS login, lost PW, and navigation [PROGRAM-45 - Improve the user interface for Saas IN PROGRESS]

Page view of PI objectives in Confluence

PI 2 Progress

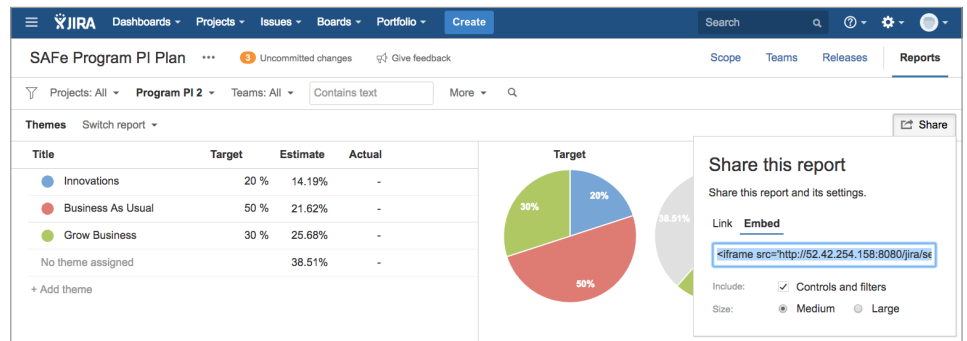
Created by John Steel, last modified about 2 hours ago

Key	T	Summary	Status	Value Stream	Theme	Release Train
PROGRAM-43	[f]	Re-design admin configuration navigation	IN PROGRESS	VS1	Grow Business	ART1
PROGRAM-30	[f]	Provide connectors to Flioffer social networks	IN PROGRESS	VS1	Innovations	ART1
PROGRAM-20	[f]	Planning for the new GreenCloud web page	CLOSED	VS1	Business as Usual	ART1
PROGRAM-16	[f]	Launch the office	IN PROGRESS	VS2	Grow Business	ART2

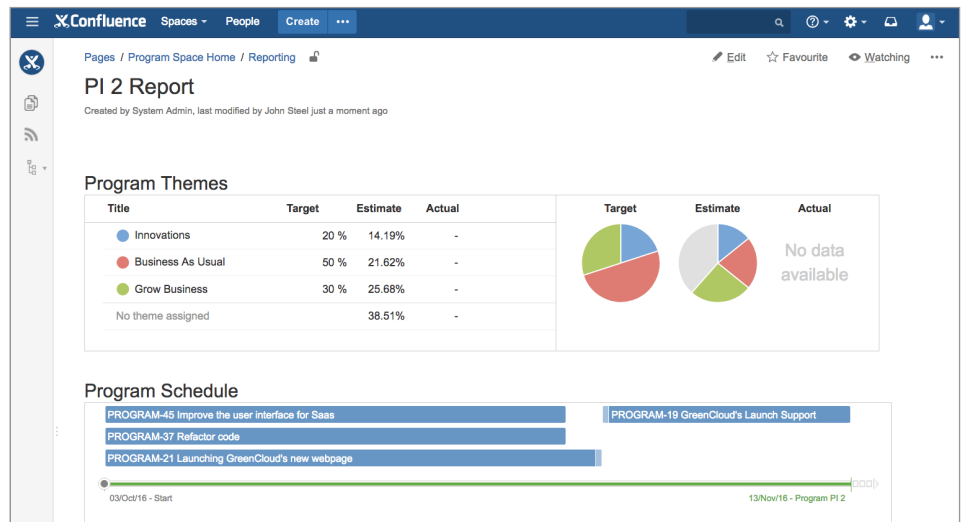
4 issues Refresh

Status report in Confluence

Portfolio for JIRA report embedded in Confluence



Multiple Portfolio for JIRA reports embedded in Confluence



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

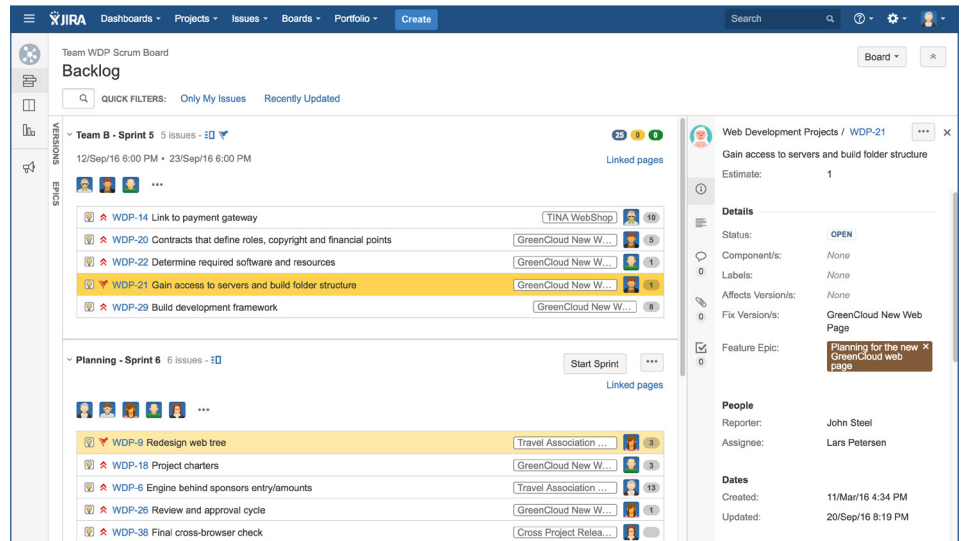
Configuration

- 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. 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 captured in JIRA Software

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

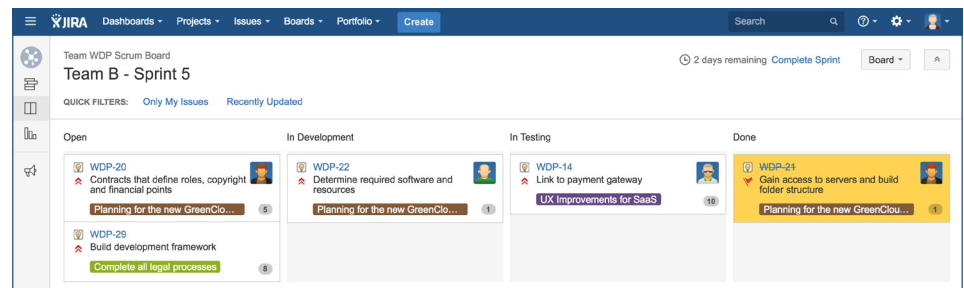
JIRA Software scrum board - active sprint and report views

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

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

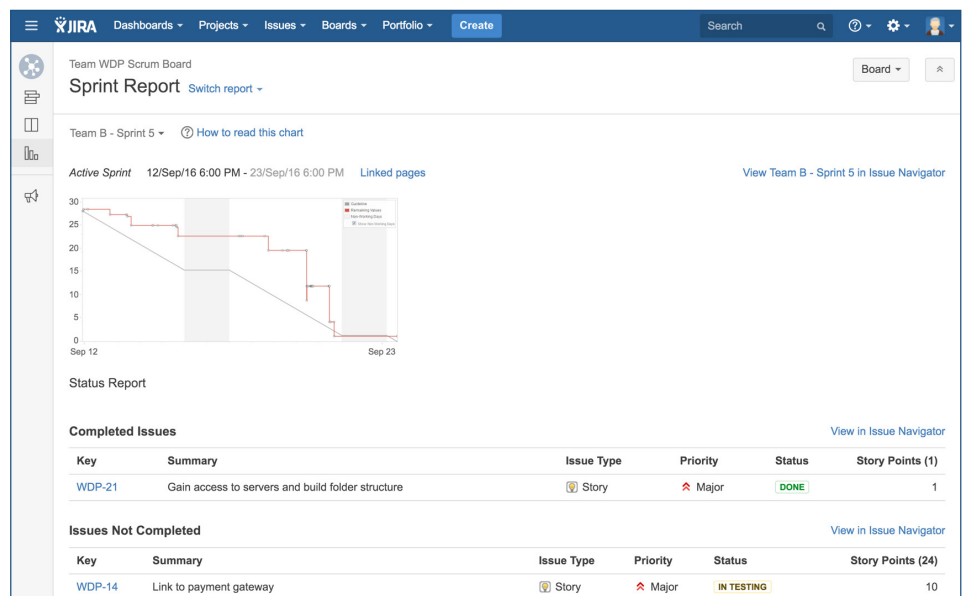
- JIRA Software scrum board
- Retrospective blueprint in Confluence

Configuration

No additional configuration required

- Use existing scrum board and Confluence configuration

Scrum board sprint report in JIRA Software



Retrospective blueprint in Confluence

Retrospective

What did we do well?

- Collaboration on development framework requirements
- Payment gateway configuration - easier than expected
- New system access process worked well and was fast

What should we have done better?

- Poor resource planning - resources not available when we need them
- Legal team unprepared for our copyright questions - should have shared sooner
- Estimation could have been better

Actions

- ☐ @Lars Petersen - Meet with legal team on upcoming requests
- ☐ @Jessie Rosewood - Challenge team on estimation during sprint planning



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® 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.



SCALING AGILE WITH ATLASSIAN AND SAFE

APPENDIX

- ¹ <http://www.scaledagileframework.com/about/>
- ² <http://www.scaledagileframework.com/portfolio-level/>
- ³ <http://www.scaledagileframework.com/program-level/>
- ⁴ <http://www.scaledagileframework.com/team-level/>
- ⁵ <http://www.scaledagileframework.com/agile-release-train/>
- ⁶ <http://www.scaledagileframework.com/program-increment/>
- ⁷ <http://scaledagileframework.com/WSJF>
- ⁸ <http://scaledagileframework.com/lean-agile-leaders>
- ⁹ <http://scaledagileframework.com/communities-of-practice/>
- ¹⁰ <http://scaledagileframework.com/program-portfolio-management/>
- ¹¹ <http://scaledagileframework.com/epic-owner/>
- ¹² <http://scaledagileframework.com/enterprise-architect/>
- ¹³ <http://scaledagileframework.com/release-train-engineer-and-value-stream-engineer/>
- ¹⁴ <http://scaledagileframework.com/product--and-solution-management/>
- ¹⁵ <http://scaledagileframework.com/system-and-solution-architect-engineering/>
- ¹⁶ <http://scaledagileframework.com/customer/>
- ¹⁷ <http://scaledagileframework.com/release-train-engineer-and-value-stream-engineer/>
- ¹⁸ <http://scaledagileframework.com/system-and-solution-architect-engineering/>
- ¹⁹ <http://scaledagileframework.com/product--and-solution-management/>
- ²⁰ <http://scaledagileframework.com/product-owner/>
- ²¹ <http://scaledagileframework.com/scrum-master/>
- ²² <http://scaledagileframework.com/agile-teams/>
- ²³ <http://scaledagileframework.com/strategic-themes/>
- ²⁴ <http://www.scaledagileframework.com/value-streams/>
- ²⁵ <http://scaledagileframework.com/agile-release-train/>
- ²⁶ <http://scaledagileframework.com/pi-planning/>
- ²⁷ <http://scaledagileframework.com/program-increment/>
- ²⁸ <http://scaledagileframework.com/solution-demo>
- ²⁹ <http://scaledagileframework.com/iteration-planning/>
- ³⁰ <http://scaledagileframework.com/iteration-execution/>
- ³¹ <http://scaledagileframework.com/system-demo/>
- ³² <http://scaledagileframework.com/iteration-retrospective/>

