



How AJ Bell made DevOps a Dodl with Atlassian tools



MO TAGARI & ANDY PASS



570%

Increase in releases per month over an 18 month journey



Agenda

Our Journey so far

Our DevOps Tool Chain

Using Automation to hit the gas pedal

Measuring Success





Co-founded by CEO Andy Bell in 1995, AJ Bell is one of the UK's largest and best regarded investment platforms.

KEY FACTS

- 398,000 customers trust us with their investments
- £75.6 billion of assets under administration
- FTSE 250 company
- Approximately 1,000 employees

Our core propositions

Flagship propositions

Established platform propositions offering a wide range of investment choice and functionality



New propositions

Digital-only platform propositions offering an easy to use, streamlined investment service



Investment solutions

A range of in-house funds and Managed Portfolio Service solutions



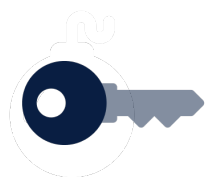
What made us start this Journey?

The business having an insatiable appetite for change and technology teams were not in a position to consistently deliver at pace with quality and stability in mind



No Alignment

4 core product lines with limited alignment of resources and ownership of releases sitting with a centralised DevOps function



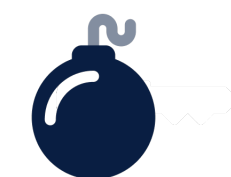
Manual Processes

Manual processes for release notes, communications and deployments



Lack of Standardization

A lack of standards across teams for development practices, tooling and pipeline automation

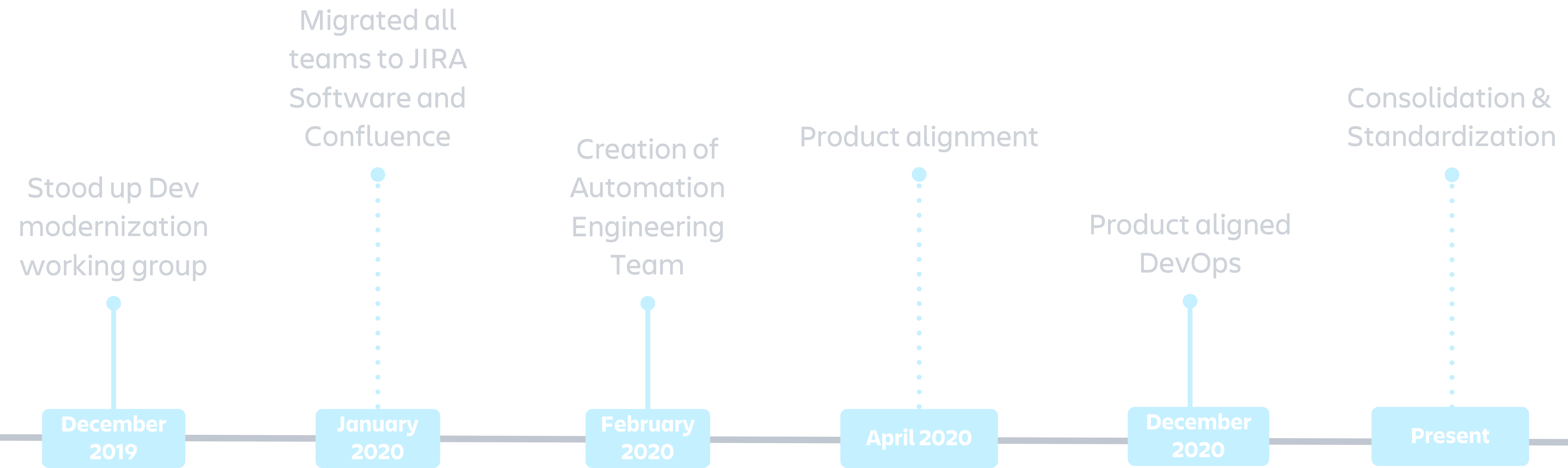


Big Bang Releases

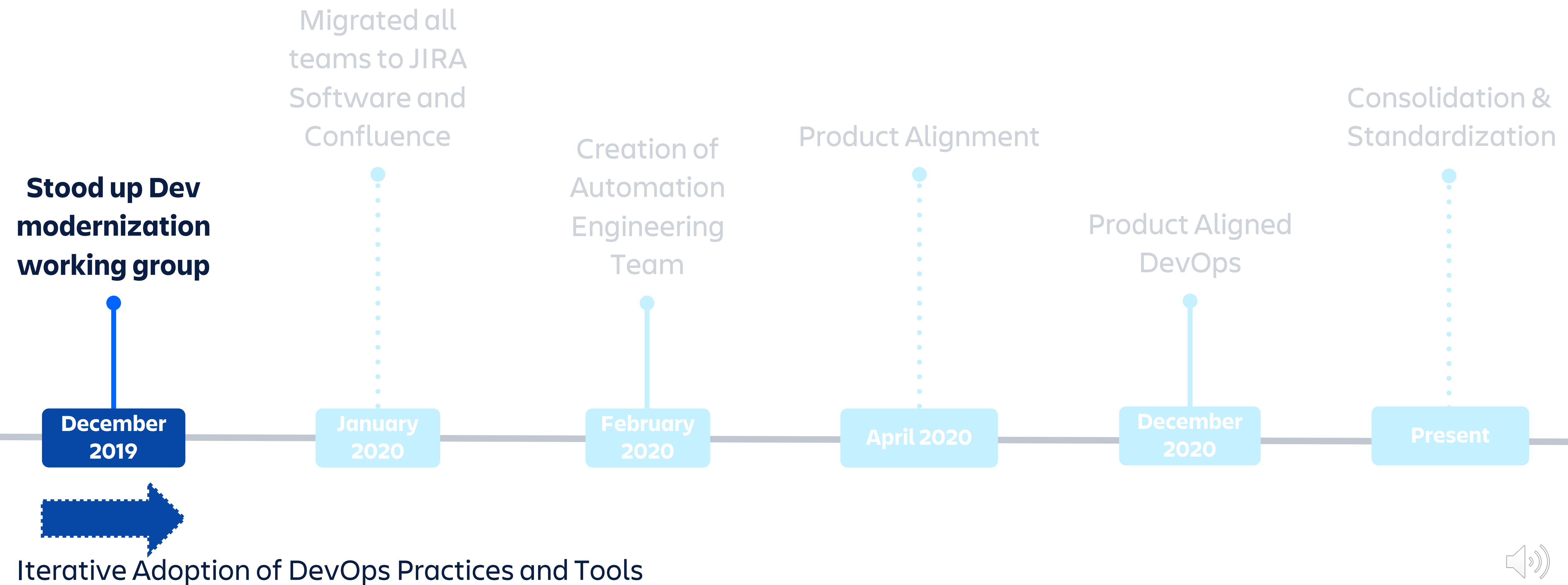
A fixed sprint release schedule to which all teams contributed leading to increased risk



OUR JOURNEY SO FAR



OUR JOURNEY SO FAR



A Singular Focus, Dev Modernization Group



Change lifecycle

Single AJ Bell SDLC standard with automated measures



Automate Governance

Control requirements embedded in toolings, logging for InfoSec and App Support, pipeline and design patterns



DevSecOps

Pipeline patterns to build, test, scan and deploy apps for CI/CD with production validation and roll backs



Reduce Risk

Quasi continuous release moving away from Sunday 11pm Big Bang



Reduce Overhead

Integration into TopDesk for streamlined and controlled change management

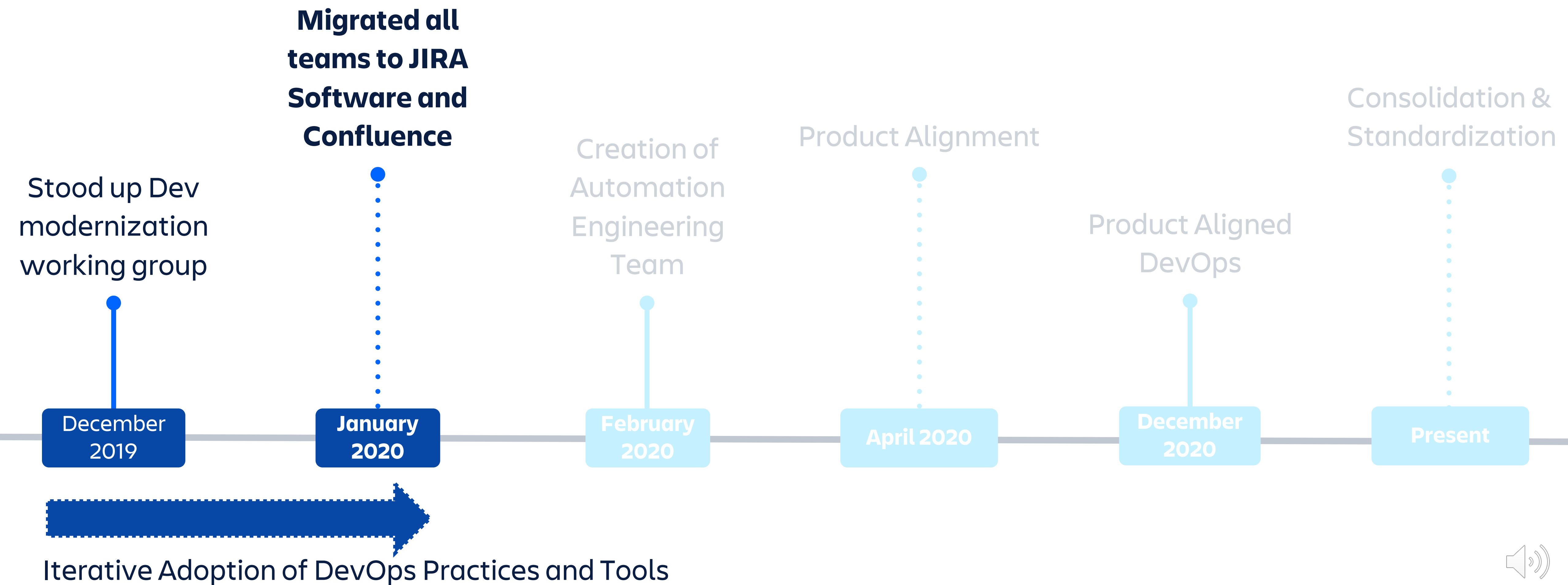


Measure Success

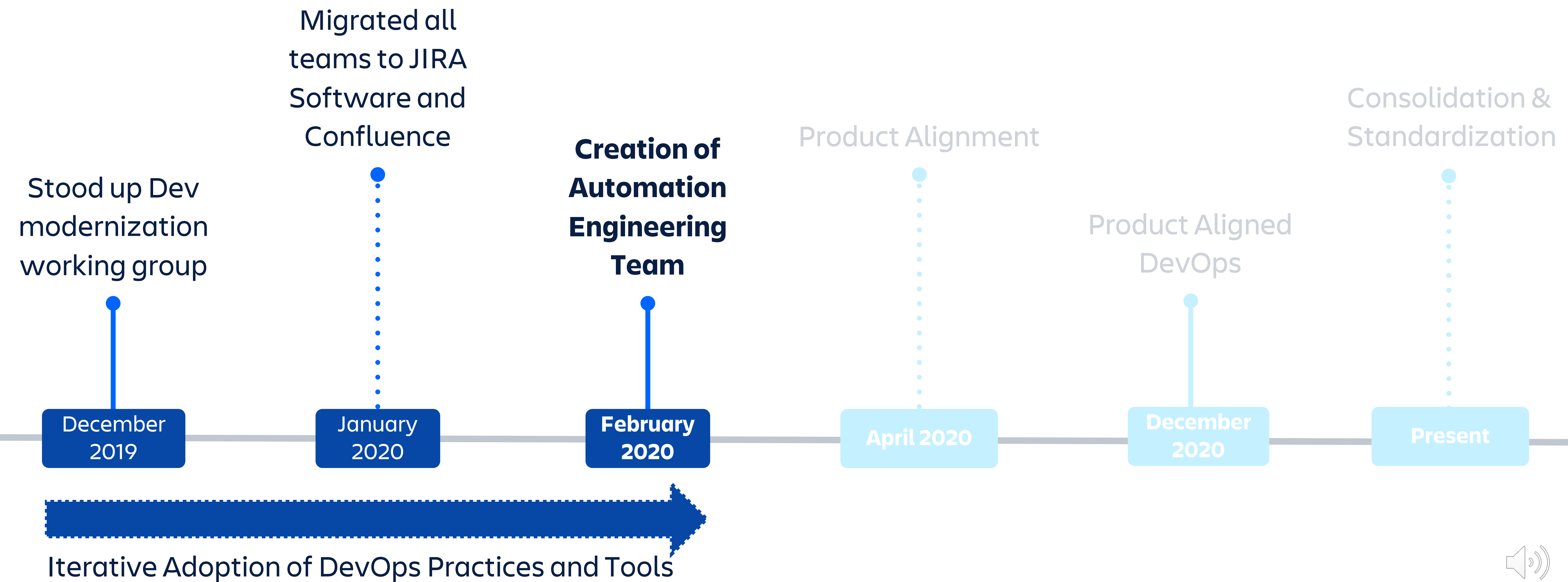
Baseline KPIs and set targets to measure adoption and effectiveness



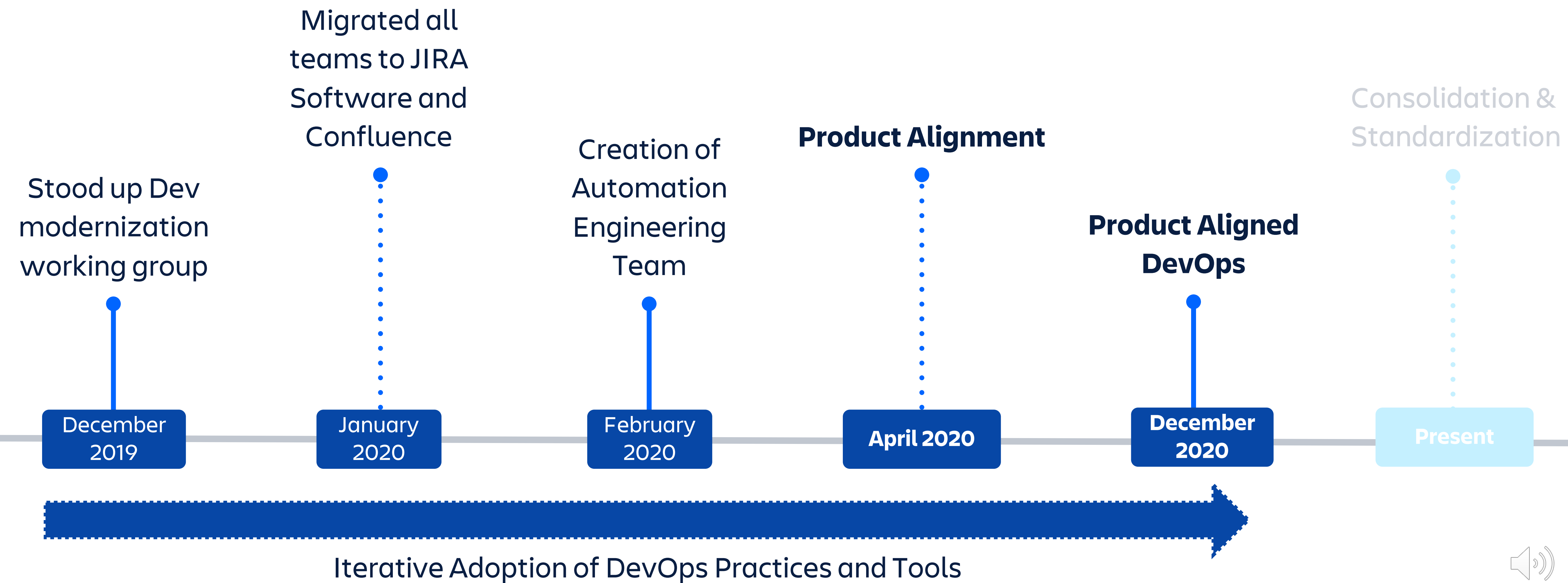
OUR JOURNEY SO FAR



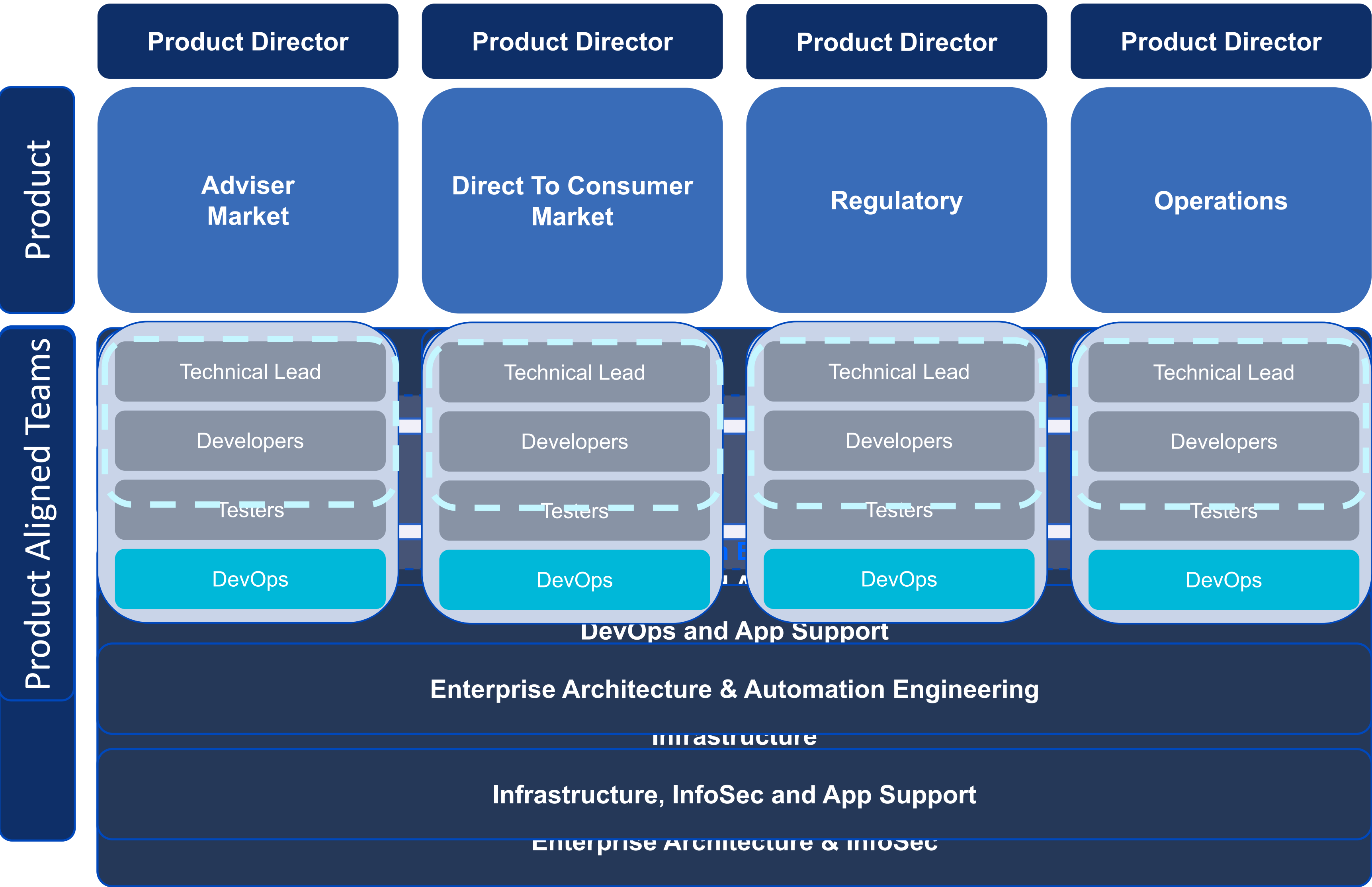
OUR JOURNEY SO FAR



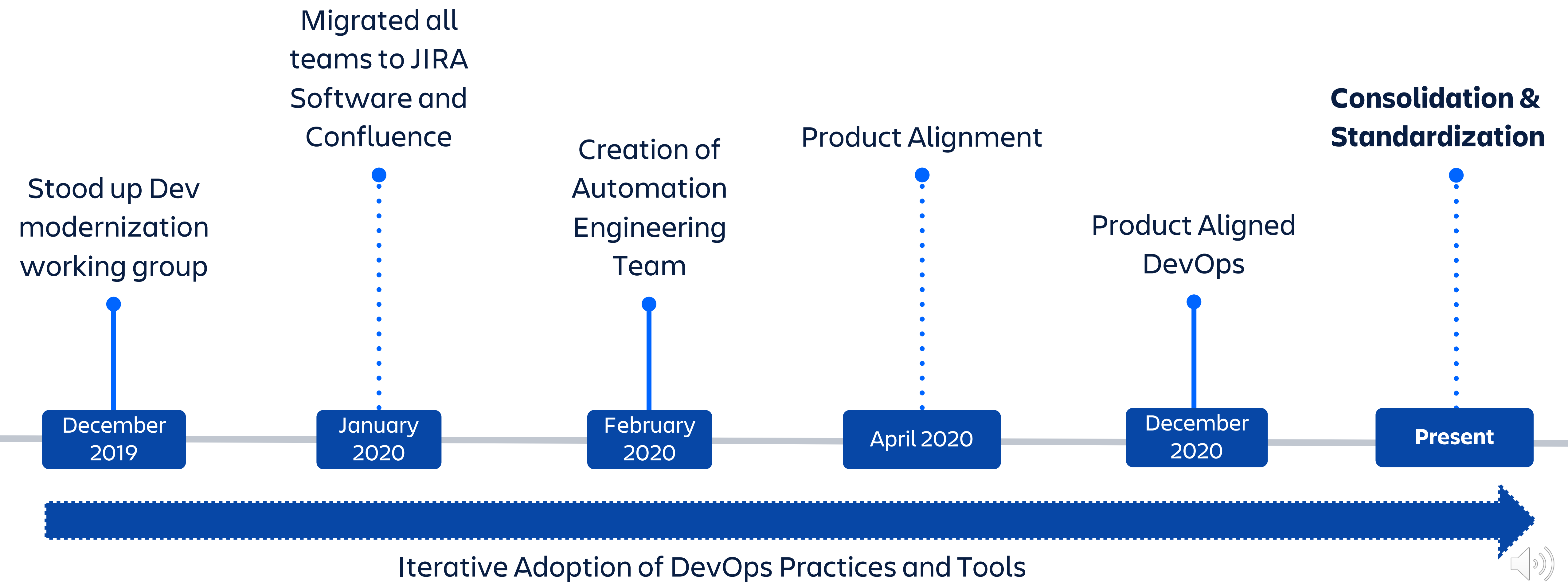
OUR JOURNEY SO FAR



Product aligned engineering
Engineering



OUR JOURNEY SO FAR



Agenda

Our Journey so far

Our DevOps Tool Chain

Using Automation to hit the gas pedal

Measuring Success



DevOps

Atlassian Tools

Atlassian's interoperability allowed us to central all our DevOps tools around one tool.

Jenkins

Using Jenkins gave us flexibility whilst also addressing security concerns. Creating a shared library helped roll out improvements quickly.

Observability

Tools such as OpsGenie, Sentry, Snyk and Graylog gave much needed visibility across our environments giving us confidence in our releases.



AJ Bell + Atlassian

“Innovation sits at the heart of AJ Bell and technology is the engine that powers that innovation. Quality and speed of delivery are crucial in ensuring we stay ahead of changing customer behaviours and continue to lead the investment platform market in the UK.”

Andy Bell
CEO



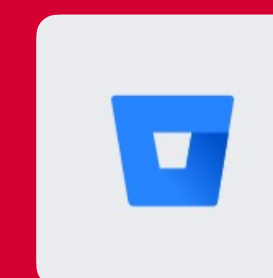
KEY CLOUD VALUE

Guaranteed uptime



Jira Software

Project and issue tracking



Bitbucket

Git code management



Confluence

Document collaboration



Atlassian Access

Security and control for cloud



Our Tools



JIRA Software

Day to day task management and the central pillar.



Confluence

Documentation and release notes



Bitbucket Cloud

Code repositories



Jenkins

Our chosen orchestration tool for deployments



Snyk

Security scanning embedded into our pipelines



SonarQube

Automated gates for code quality

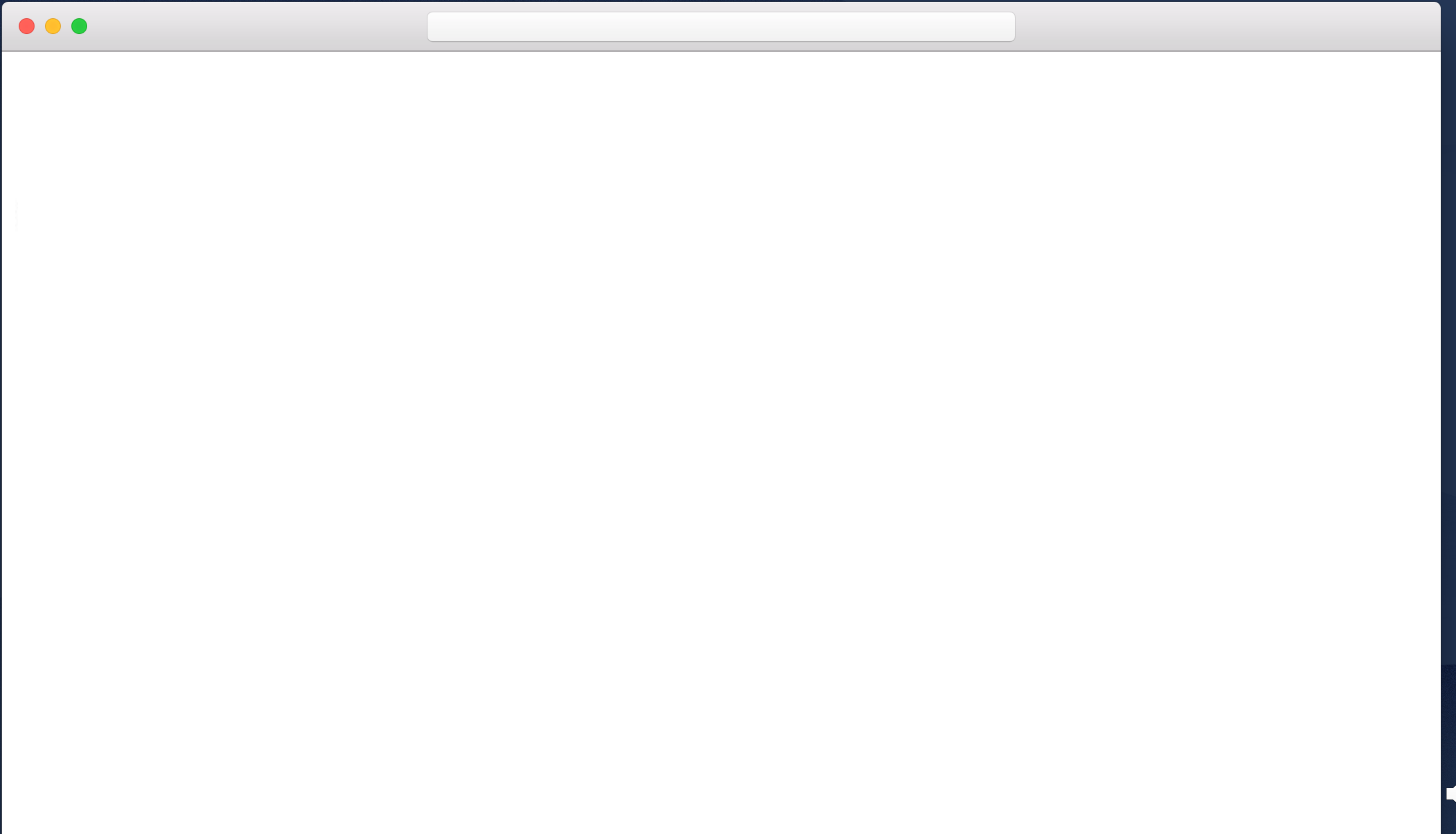


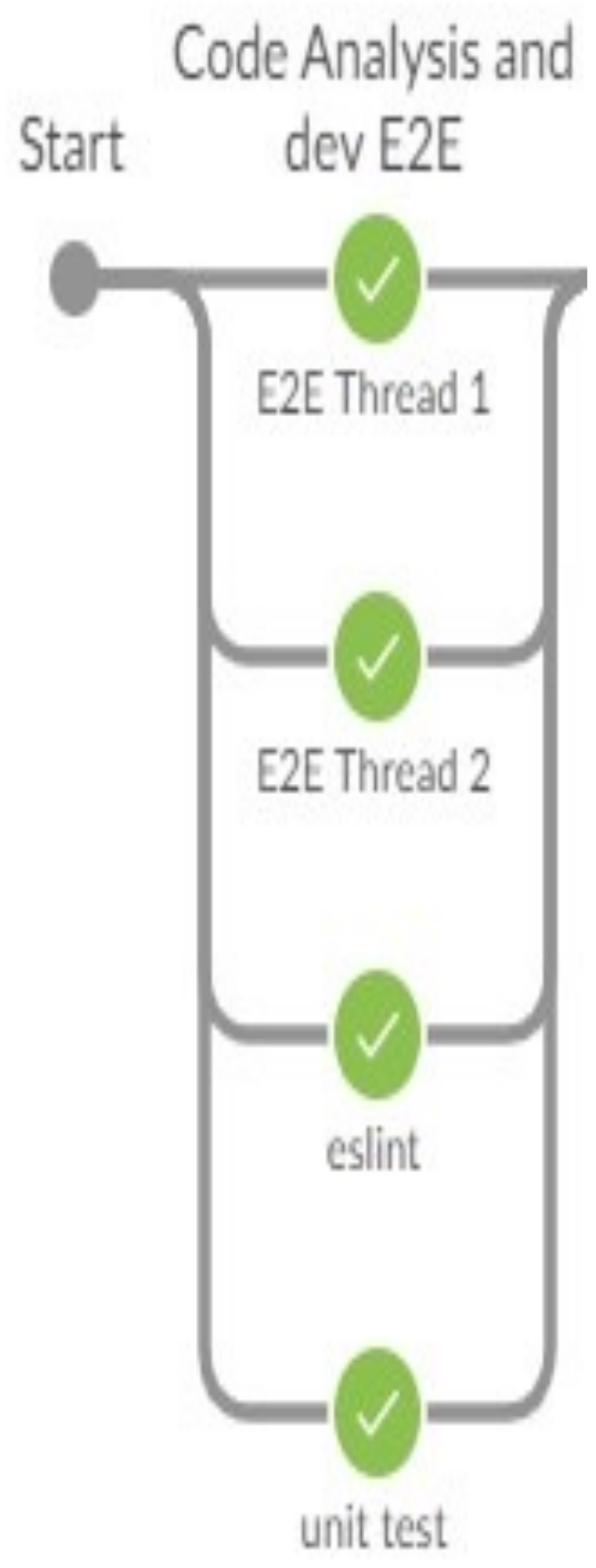
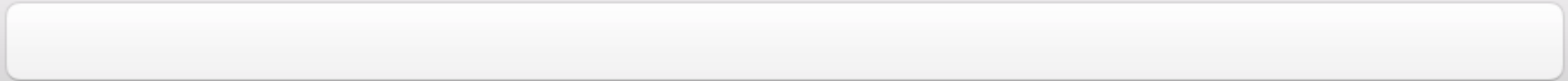
What did we do to improve?

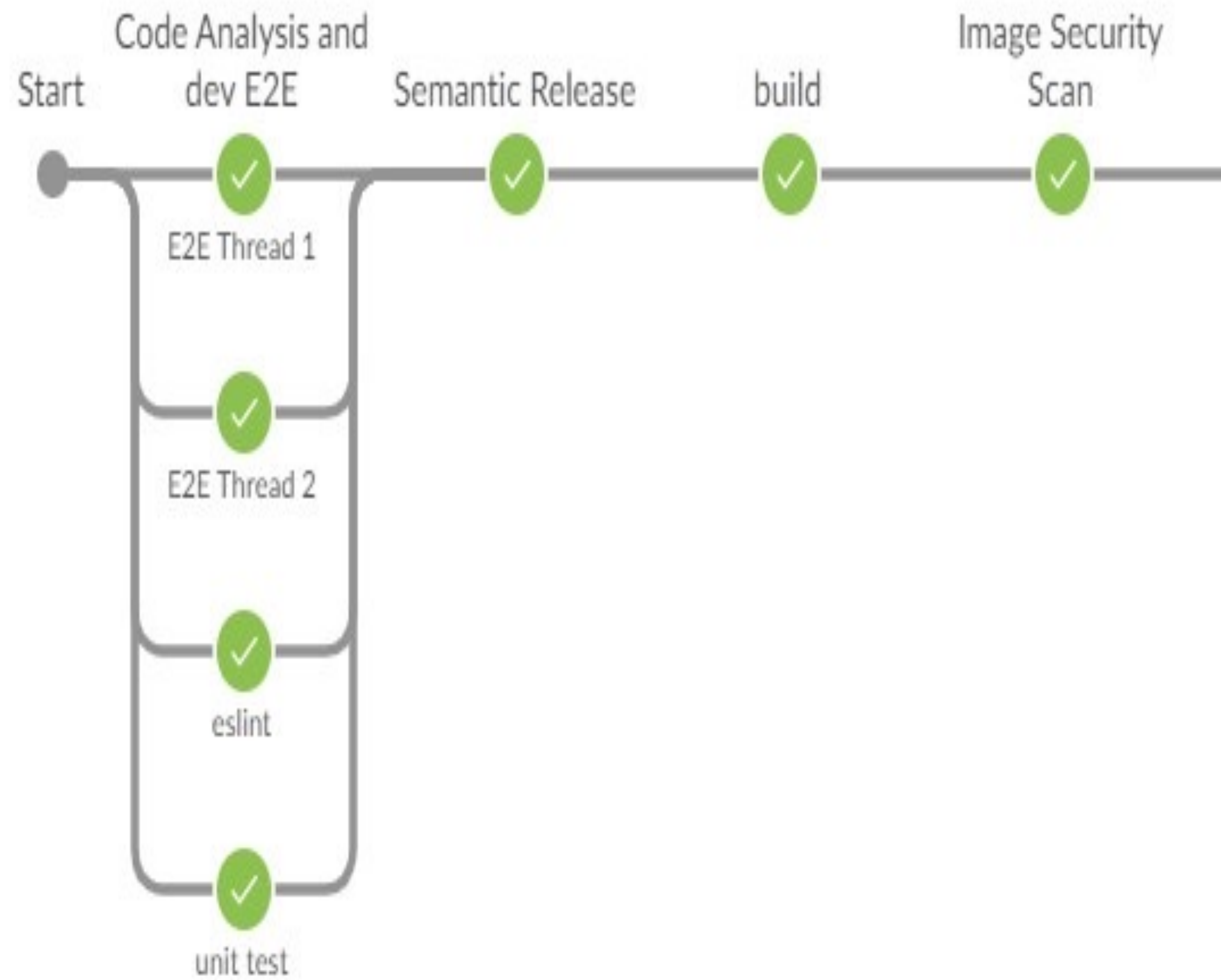
Through a collaborative process with our developers, testers and DevOps engineers at its heart we were able to re-engineer our DevOps tool chain and re-align our change lifecycle to support this.

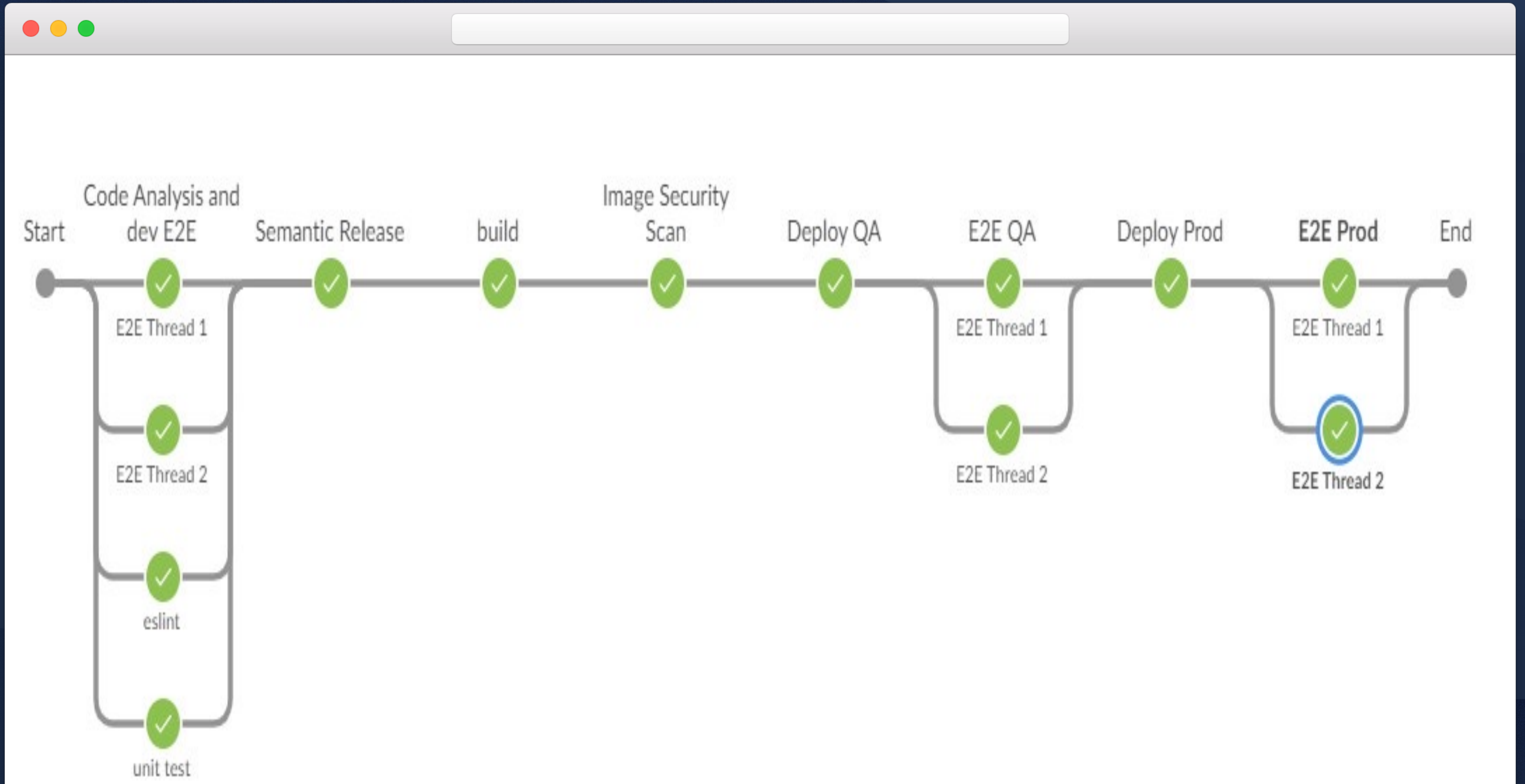
- We standardized where possible and set up shared libraries to enable fast changes globally to JenkinsFiles in team repositories.
- Moving away from release branches and to trunk based development helped to facilitate more frequent releases and ultimately eliminated the need for release notes.
- Taking advantage of Jira Software's ability to communicate to different tools meant we were able to gain visibility by integrating Jenkins and Bitbucket with JIRA
- Introducing quality gates and shifting testing left using SonarQube, automated testing and Snyk.











INTEGRATING JENKINS - JIRA

Visibility

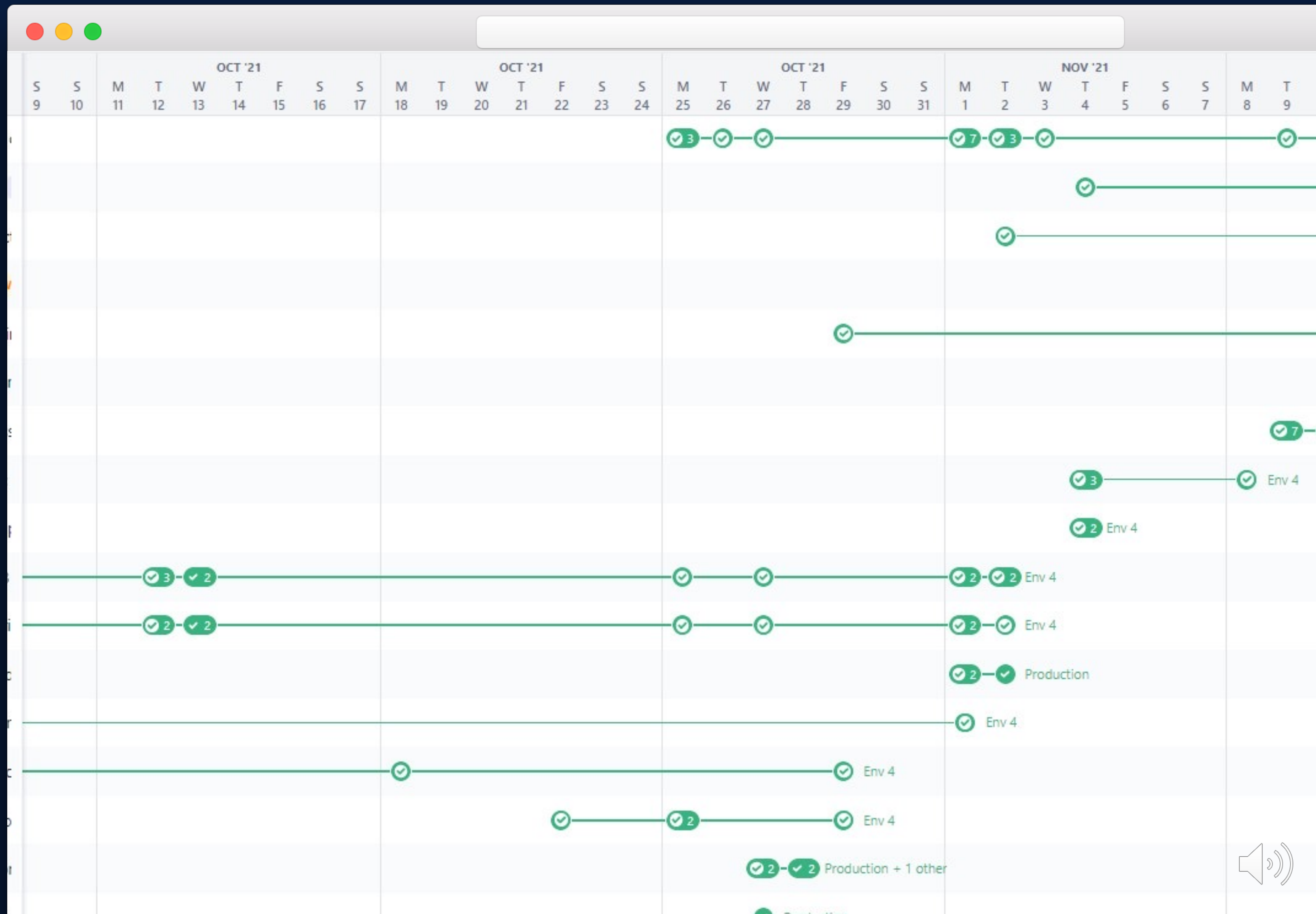
Using the Jira Plug in for Jenkins enables us to send data on deployments across environments back to JIRA.

Reporting

Connecting Jenkins and JIRA enables us to get easy information on cycle time from commit to production.

Automation

Jenkin's moves tickets into QA only once the ticket has been fully deployed into our test environment.



Agenda

Our Journey so far

Our DevOps Tool Chain

Using JIRA Automation to hit the gas pedal

Measuring Success



Why Automate? Why not?

Increased Pace of Change

Take away the pain of change by automating as much as possible to help encourage teams to release frequently.

Better Visibility

Gain more insight across your tools by utilising JIRA Automation's ability to send API calls.

Reduce Risk, Ease Compliance

Take the heavy lifting out of releasing and ensure that it's easy to link a ticket to a code change and subsequent deployment.



What did we do to improve?



Integrated JIRA and Bitbucket

Reducing daily overhead on developers by connecting Bitbucket and JIRA.



Gained Deployment Visibility

Utilised the functionality between Jenkins and JIRA to get more insight into what is deployed where and when



Reducing Release Admin

Connection to topDesk / Change MGT and Release Notes



Automated our Governance

Utilising Jira and Confluence's REST API to create templates and check adherence to key processes



ATLASSIANS AUTOMATION

Conditions

These can be used to give a lot of flexibility and really narrow down the scope of a rule. Use them wisely!

Web Requests

API calls can be used to link into pretty much any tool in your stack and close the gap between tools.

Variables

Chain together API calls and save data from the JSON response in variables to be used later in the rule.

The screenshot displays the Atlassian Automation interface. At the top, the title 'Automation' is followed by a green 'ENABLED' status tag. Below this, the rule is named 'Release creation - AJBYI'. A sidebar on the left contains links for 'Rule details' (with an actor note: 'Actor: User who triggered the event') and 'Audit log'. The main workflow area shows a sequence of steps: 1. 'When: Version created' (trigger event). 2. 'If: Compare two values' (conditional step, marked with a yellow icon and a close button). The condition is 'Checks if: {{version.name}} does not contain UC -'. 3. 'And: Compare two values' (conditional step, marked with a yellow icon). The condition is 'Checks if: {{version.name}} does not contain CAB -'. 4. 'Then: Send web request' (action step, marked with a blue icon). 5. 'And: Create variable' (action step, marked with a blue icon). The action is 'Create smart value variable {{topDeskNum}}'. On the right side, a detailed view of the 'Advanced compare condition' is shown. It includes a description: 'Compares two values using smart values and regular expressions. Learn more about iss Advanced compare condition.' It has input fields for 'First value *' containing '{{version.name}}', 'Condition' set to 'does not contain', and 'Second value' containing 'UC -'. A link '> What values can I compare?' is at the bottom.

Automation ENABLED

Release creation - AJBYI

- Rule details
Actor: User who triggered the event
- Audit log

When: Version created

If: Compare two values ×
Checks if:
{{version.name}} does not contain UC -

And: Compare two values
Checks if:
{{version.name}} does not contain CAB -

Then: Send web request

And: Create variable
Create smart value variable
{{topDeskNum}}

Advanced compare condition 🗑

Compares two values using smart values and regular expressions. [Learn more about iss Advanced compare condition.](#)

First value *
{{version.name}}

Condition
does not contain

Second value
UC -

> What values can I compare?



How we approached Governance

Automate where possible

Look to automate as much as possible, if automation isn't possible then think outside the box.

Full visibility

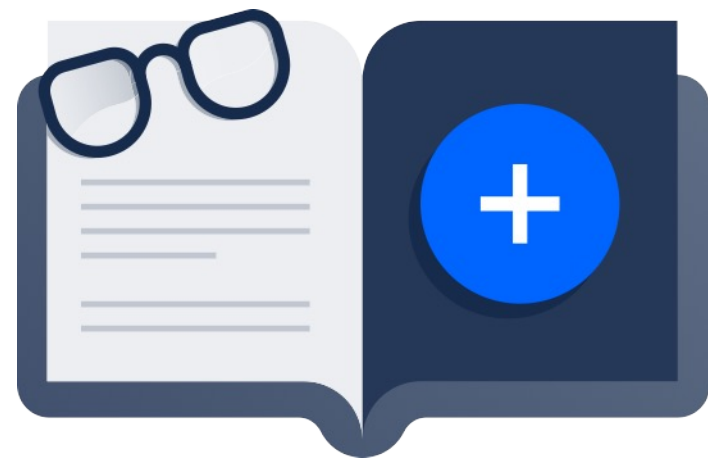
Take advantage of JIRA Software's connectivity, by connecting to tools across your tool chain it's easy to see what's happening.

Make it easy

Where manual processes are required keep them easy to follow and easy to complete.



Addressing Governance



Automated Reminders

Help Product Owners to ensure key steps aren't missed.



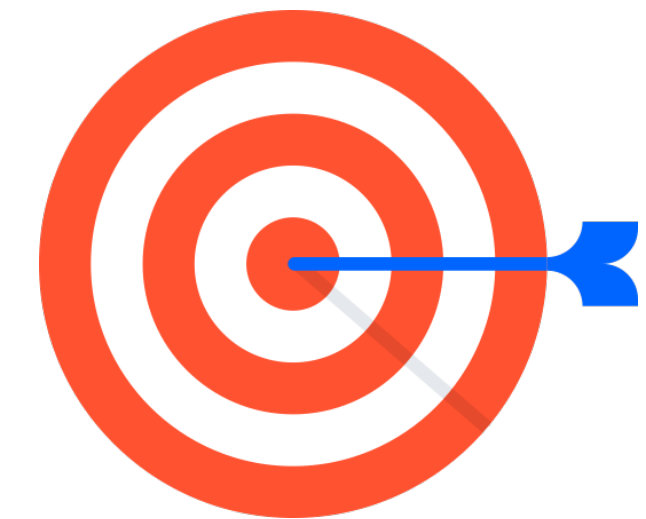
Confluence Templates

Custom templates for major releases ensuring standardisation.



Improved Auditability

Connecting Jira to Bitbucket, Jenkins and TopDesk ensures a full E2E picture.



Guardrails not Gates

These enable governance without impeding development.



Agenda

Our Journey so far

Our DevOps Tool Chain

Using Automation to hit the gas pedal

Measuring Success



Always measure against KPIs

Set baselines early

Make sure you can measure your success knowing where you have come from and where you are heading

Driving Adoption, Measuring Success

As much as possible take assumptions away by having solid data validating decisions and pivot where needed

Make it visible, No Secrets

Share the metrics organisation wide as teams seeing where they need to improve will drive focus and collaboration



EAZYBI, JIRA AND CONFLUENCE

Easily Accessible

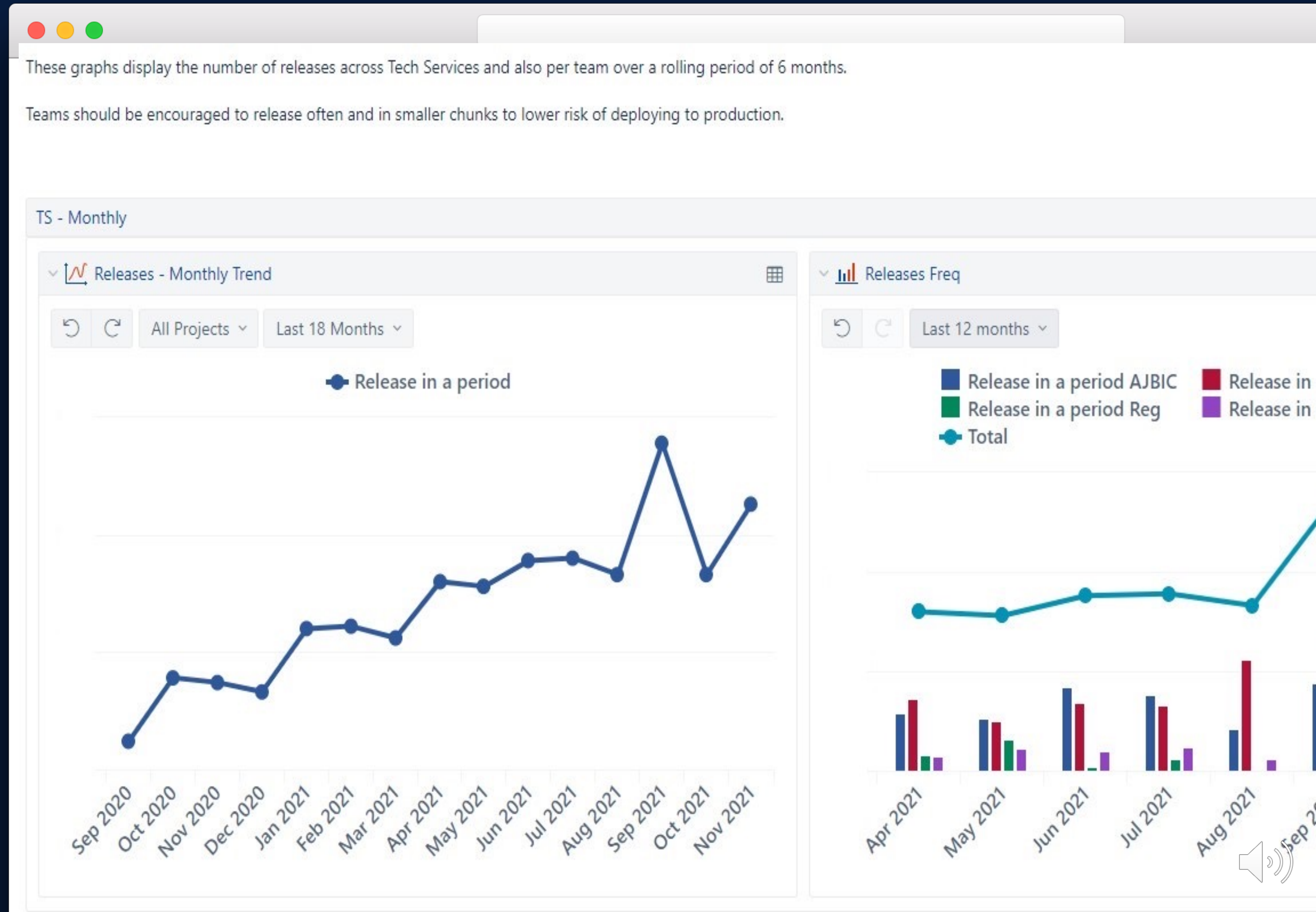
Utilising EazyBI and Confluence enabled us to build out a self service reporting suite in Confluence.

Track Progress

By measuring key metrics such as releases, bugs, production incidents and time to recover

Use the Data

Reports are useless unless you challenge the findings and derive learnings to continually improve





Thank you!

MO TAGARI, ANDREW PASS | AJ BELL |

[HTTPS://WWW.AJBELL.CO.UK/CAREERS](https://www.ajbell.co.uk/careers)

[HTTPS://WWW.LINKEDIN.COM/COMPANY/AJBELL/LIFE/OURTECHTEAM/](https://www.linkedin.com/company/ajbell/life/ourtechteam/)

