



What's this DevEx and Platform Engineering I keep hearing about?

Bob Walker | Field CTO | Octopus Deploy

@bjwalker

bob.walker@octopus.com

@bjwalker

bobwalker.octopus.app

Thank You TechBash 2024 Sponsors!

Visit <https://techbash.com/#Sponsors>
to visit our sponsors' websites and
learn more about them!



Bob Walker



Field CTO
Octonaut since 2018

 @bobjwalker

 bob.walker@octopus.com

 @bobjwalker

 bobjwalker.octopus.app





**Intro session for anyone
curious about DevEx
and Platform Engineering**

Agenda

- Intro to Developer Experience
- Intro to Platform Engineering
- Paved vs. Golden Paths
- Why now?
- Avoiding Pitfalls
- Measuring success
- Resources

Intro to Developer Experience (DevEx)



Multi-Day Effort to Merge to Main



Sarah Ashri 10:28

Update on my [card](#): my e2e tests are failing due to flaky tests . I'll keep pushing them and will merge them and close the card once all the tests pass



Sarah Ashri 09:02

replied to a thread: [Remove UpdateInstanceResourcesOnTaskCapChange Feature...](#)

Merged after 12 e2e tests and 3 branches ... now to the 2nd one



3



#9.12403.0-rm-task-cap-flag-2-3	sarah/rm-task-cap-flag-2-3	Tests passed: 61, ignored: 12	sarah ashri <sarah.ash...4	nautilus-linux-1759467	9 Apr 24 07:42	1h 15m 40s	⋮	⋮
#9.12391.0-rm-task-cap-flag-2-2	sarah/rm-task-cap-flag-2-2	Tests failed: 3, passed: 58, ignored: 12	sarah ashri <sarah.ash...1	nautilus-linux-1756183	8 Apr 24 14:31	1h 56m 16s	⋮	⋮
#9.12392.0-rm-task-cap-flag-2-2	sarah/rm-task-cap-flag-2-2	Tests failed: 72 (72 new), passed: 0, muted: 1	No changes	nautilus-linux-1756009	8 Apr 24 14:18	12m 37s	⋮	⋮
#9.12378.0-rm-task-cap-flag-2-2	sarah/rm-task-cap-flag-2-2	Tests passed: 61, ignored: 12	sarah ashri <sarah.ash...3	nautilus-linux-1754102	8 Apr 24 08:38	1h 09m 39s	⋮	⋮
#9.12390.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 72 (23 new), passed: 0, muted: 1	sarah ashri <sarah.ash...1	nautilus-linux-1757154	8 Apr 24 16:00	15m 33s	⋮	⋮
#9.12389.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 38 (37 new), passed: 23, ignored: 12	No changes	nautilus-linux-1755091	8 Apr 24 14:18	1h 42m 30s	⋮	⋮
#9.12381.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 1 (1 new), passed: 60, ignored: 12	sarah ashri <sarah.ash...1	nautilus-linux-1755054	8 Apr 24 10:55	1h 17m 39s	⋮	⋮
#9.12377.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 3 (2 new), passed: 58, ignored: 12	No changes	nautilus-linux-1753759	8 Apr 24 07:07	58m 34s	⋮	⋮
#9.12360.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 6, passed: 55, ignored: 12	No changes	nautilus-linux-1747029	5 Apr 24 16:11	1h 56m 53s	⋮	⋮
#9.12360.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 72 (69 new), passed: 0, muted: 1	No changes	nautilus-linux-1745694	5 Apr 24 14:56	23m 26s	⋮	⋮
#9.12354.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 3 (3 new), passed: 58, ignored: 12	sarah ashri <sarah.ash...1	nautilus-linux-1743459	5 Apr 24 10:42	1h 25m 44s	⋮	⋮
#9.12342.0-rm-task-cap-flag-2	sarah/rm-task-cap-flag-2	Tests failed: 2, passed: 59, ignored: 12	sarah ashri <sarah.ash...1	nautilus-linux-1740461	4 Apr 24 17:14	1h 11m 22s	⋮	⋮

Another Term: Yak Shaving

**THE TERM YAK SHAVING REFERS TO THE
SEEMINGLY ENDLESS SERIES OF SMALL
TASKS THAT HAVE
TO GET DONE BEFORE
A PROJECT MOVES
FORWARD.**



Developer Experience

The systems, technology, process, and culture that influence the effectiveness of software development

Developer Experience

It gives a name to something senior people and leaders have been partially focusing on for years

Developer Experience

You can tell a lot by a company based on
*the onboarding experience and
how hard it is to get to production.*

Example: Automating developer machine setup



Automating developer machine setup with Chocolatey



Bob Walker

August 26, 2019 • 8 mins

DevEx People Impact

- **Productivity:** how quickly or simply a change can be made to a codebase
- **Impact:** how frictionless it is to move from idea to production
- **Satisfaction:** how the environment, workflows, and tools affect developer happiness

Tangible Impact

- New people
 - The F5 Contract
 - Time to first commit
- Feature work
 - Lead time for changes
 - Deployment frequency
 - Change failure rate
 - Mean time to recovery
- Engineering NPS

Tangible Impact

- New people
 - The F5 Contract
 - Time to first commit
- Feature work
 - Lead time for changes
 - Deployment frequency
 - Change failure rate
 - Mean time to recovery
- Engineering NPS



Ideal DORA Metrics

- Lead Time for Changes (commit to prod)
- Deployment Frequency (to production)
- Change fail percentage (failures in production)
- Time to recovery (fix changes in production)

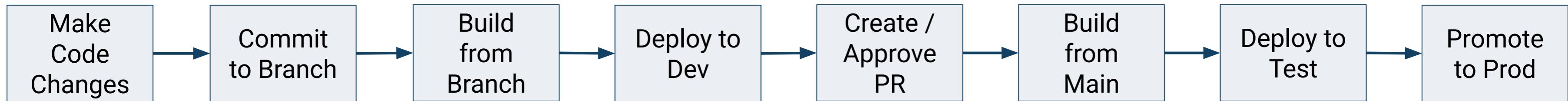
Yak Shaving DORA Metrics

- Lead Time for Changes (commit to prod)
- Deployment Frequency (to production)
- Change fail percentage (failures in production)
- Time to recovery (fix changes in production)

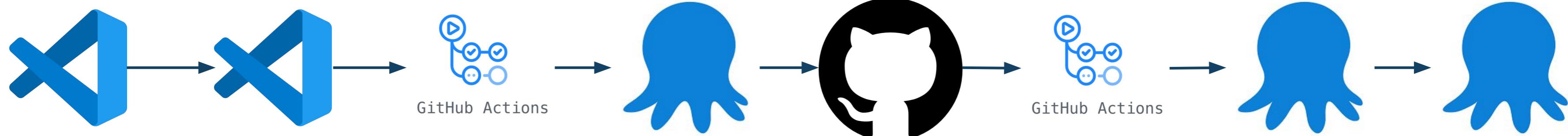
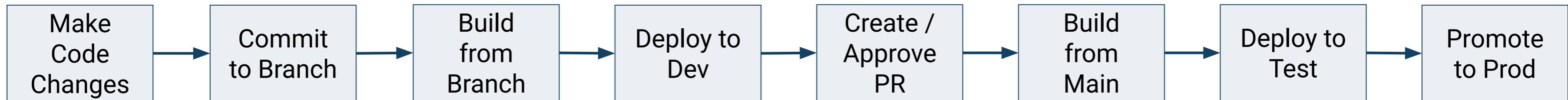
Developer Experience

Saving 15 minutes a day
adds up to 60 hours a year

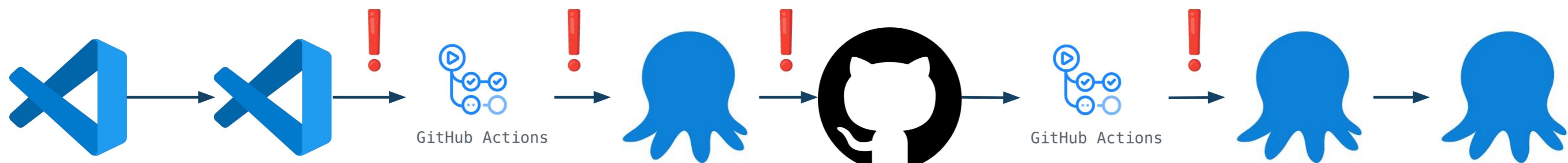
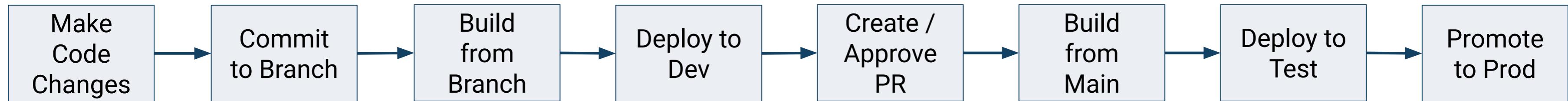
Typical Workflow



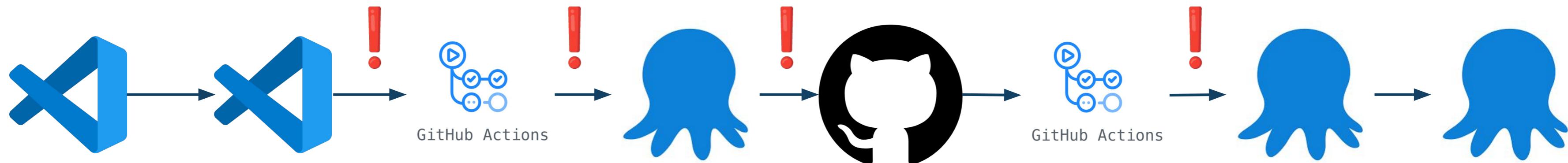
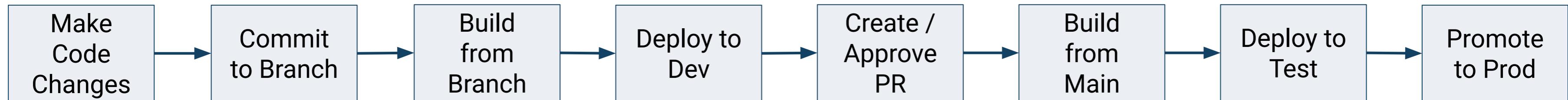
Typical Workflow



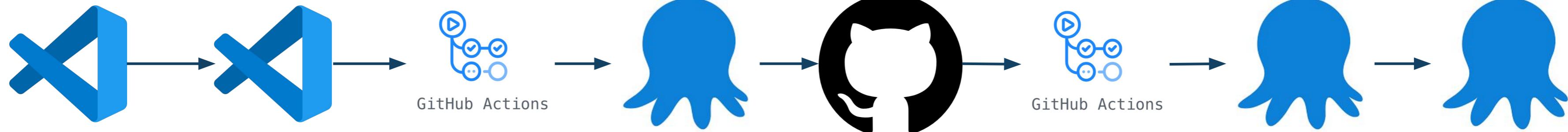
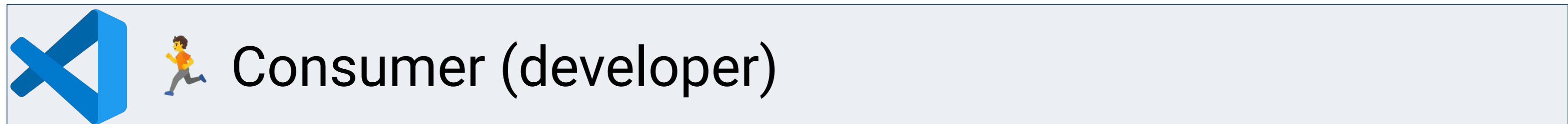
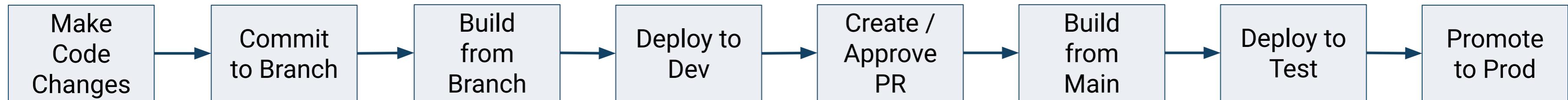
Typical Workflow



Typical Workflow



Typical Workflow



The result: frustrating experience jumping between tooling



DevEx is the responsibility of multiple people and teams



ReactionGIFS.me

Internally at Octopus

- **R&D Leadership Team (RLT):** The senior people in R&D
- **Individual Teams:** Improving the developer experience for their corner of Octopus.
- **Foundations Groups:** Make Octopus simpler and safer to make changes to
 - Backend Foundations - recently migrated from homegrown ORM to EF
 - Front-end Foundations - worked the UI to simplify it and make it easier to change
 - Build and Delivery - working on autoscaling build agents

Developer Experience

**It's not just automation, some of it can be solved
by better documentation**

New People

Getting started

- 👉 Make sure you've read the [Contributing](#) guide before continuing.
- 👉 Please make some noise in [#team-backend-foundations-requests](#) if something in this guide isn't working correctly.
- 🤓 If you discover a problem please update the guide to help the next person!
- 💻 Want to edit one of the images in this guide? [Click here](#).

Welcome to Octopus Deploy! We're excited to have you join us. This guide will help you get set up and oriented with Octopus Deploy, and the development environment. If we have done a good job, you will achieve all these goals in under two hours.

1. You will run the latest public release of Octopus Server with a single command then create and run an operations runbook
2. You will orient yourself with the development environment by querying the HTTP API and SQL Database, inspecting some logs and other files, and making a code change
3. You will have a fully capable development environment
4. You will make some simple code changes and see their effects

In this first part of the guide, you will install the necessary tools and set up your development environment.

Pre-requisites

1. Ensure you're operating system is up to date.
2. Install [git](#) and authenticate with GitHub.
3. Install [Docker Desktop](#).
 - Due to licensing of Docker Desktop, please make sure you follow the [Docker Hub guide in Confluence](#) to ensure you have the correct license.
4. Install the latest version of the [.NET SDK](#).
5. Install the latest LTS version of [NodeJS](#).
6. Install [pnpm](#).

Feel free to install these using your systems package manager, or by downloading them from the links provided. Now would also be a good time to download and other tools you'd like to use such as git clients, IDEs, etc.

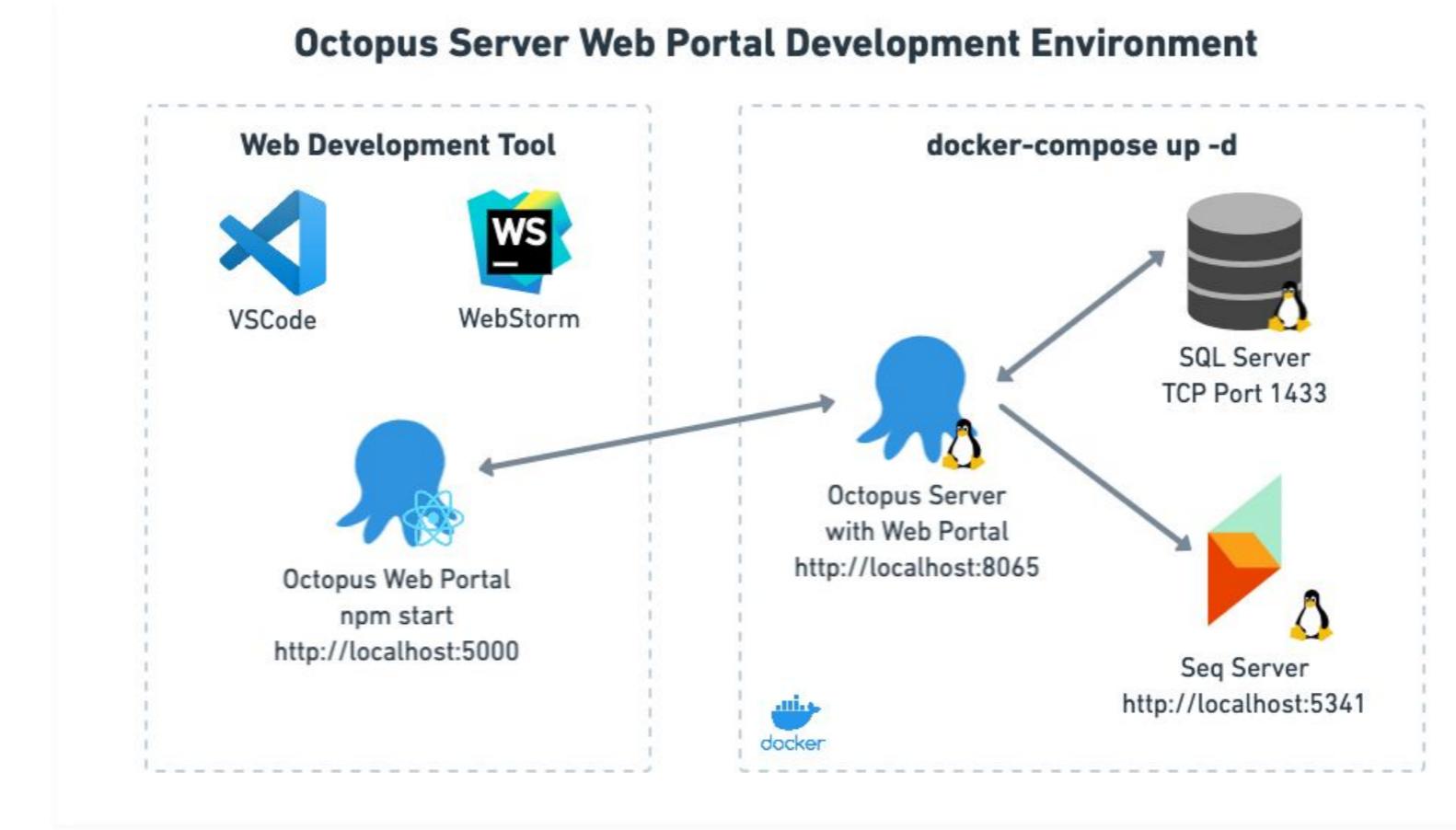
Platform Support

If you have a new developer asking which platform they should use, are considering switching platforms, or otherwise want guidance on how to deal with a platform difference, please read [Developer Platform Choices and Support](#).

The F5 Contract

Octopus Web Portal Development Environment

This section is where we will run the Octopus Server web application in a hybrid native/docker mode and start making some code changes and seeing their effects.



This guide should be enough for you to get started with the Octopus Web Portal development environment. For a more in-depth guide, see the [Octopus Web Portal development guide](#).

We will start with web application development so you can make some code changes and see them take effect. We will install `npm`, make some simple code changes, and see their effect.

Step 1: Compose up the Octopus Server development stack

These steps will stand up the Octopus Server stack using the code on your computer from GitHub. In the next steps, you will stop the service you want to develop and debug, leaving the rest running as-is.

1. Compose up the Octopus Server development stack using the `docker-compose.yml` file.

```
docker compose up -d
```

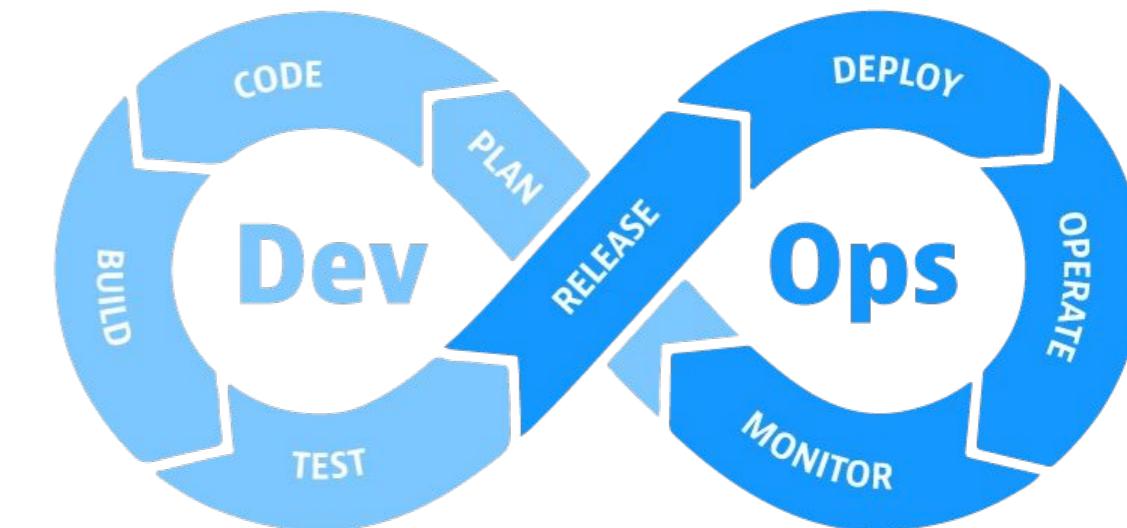
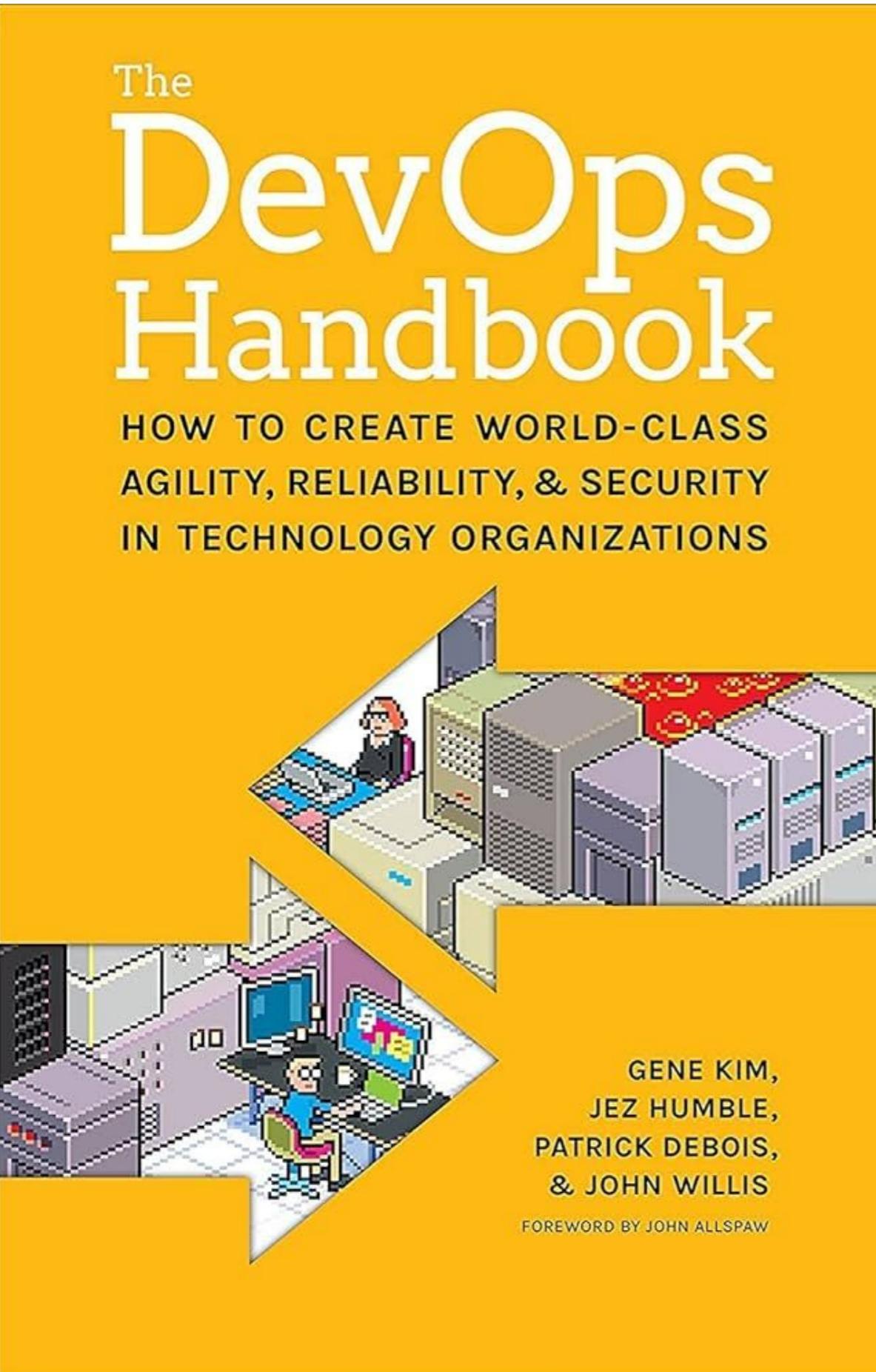


Step 2: Build and run the Octopus Server web portal

Intro to Platform Engineering



DevOps has been really successful!



Three Stages of DevOps

- **Divided:** Pre-DevOps, there are separate silos for development and operations with conflicting goals
- **Aligned:** Aligned Goals and objectives to remove conflict between teams
- **Combined:** Removing metaphorical walls allows cross-functional teams to build and run software



Combined Stage

You build it, you run it

Platform Engineering

Helps you get the benefits of
DevOps at Scale

Platform Engineering

**Build processes to manage infrastructure,
software delivery, and operation tasks.**

Platform Engineering

Helps improve the developer experience.

Platform Engineering is not the only way to improve DevEx

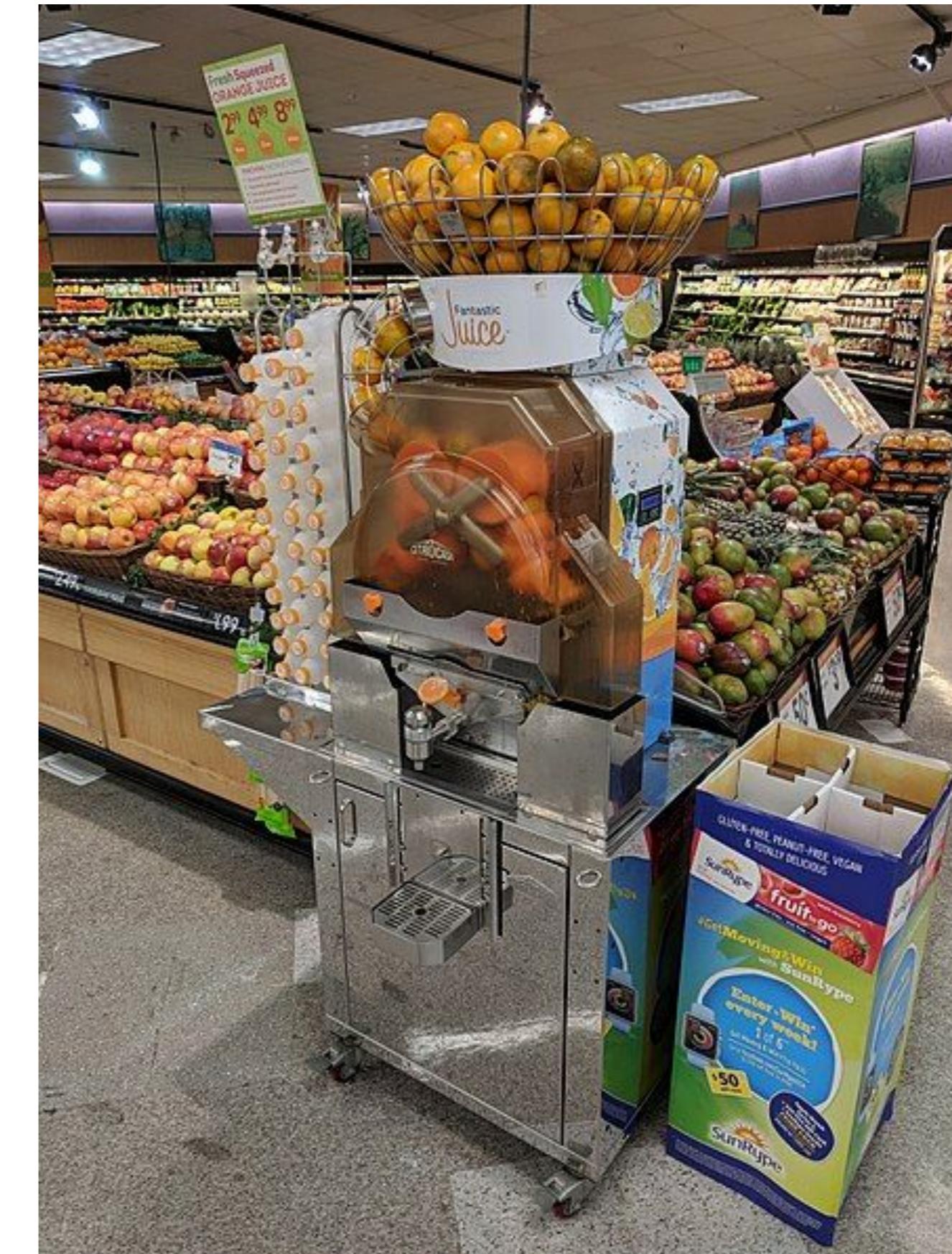


Paved vs. Golden Paths



Paved Path

- Looks to improve an existing process.
- Attempts to leverage automation to remove a specific pain point.
- May or may not solve the underlying problem.
- May or may not be better than current processes.



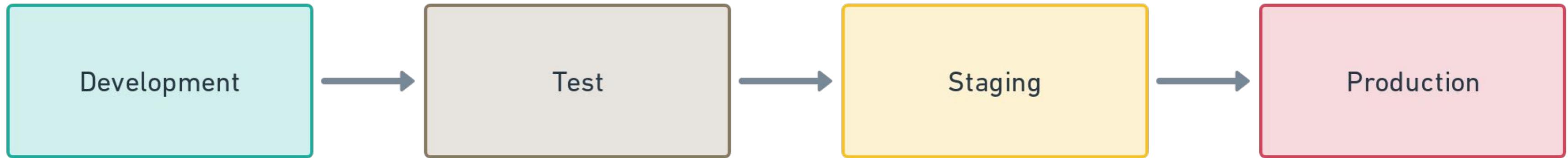
Golden Path

- Reduce burden and remove pain points.
- Solves specific problems developers are encountering.
- May involve automation, or just a new way of doing something.
- Better than what developers currently do, and is easy to adopt.
- Not forced onto a team, it is too enticing to not use it.

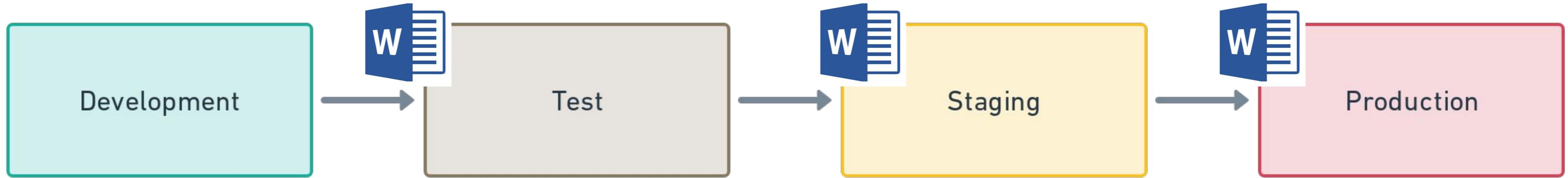


Image ID: 2HXHAK2
www.alamy.com

Paved Path Example



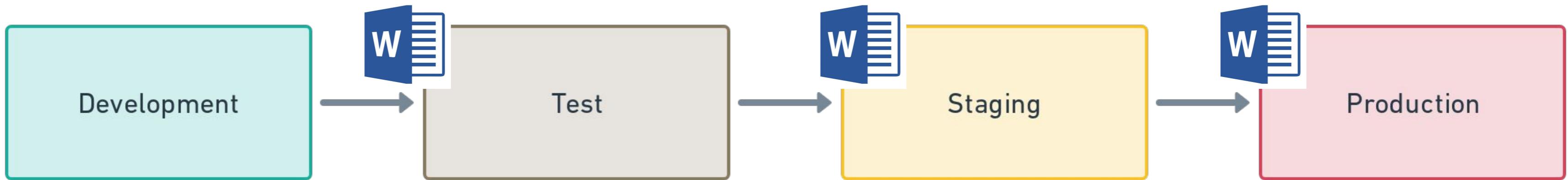
Paved Path Example



Paved Path Example



WORD FILES ARE JUST FANCY XML FILES



Paved Path Example



WORD FILES ARE JUST FANCY XML FILES



Why Now?

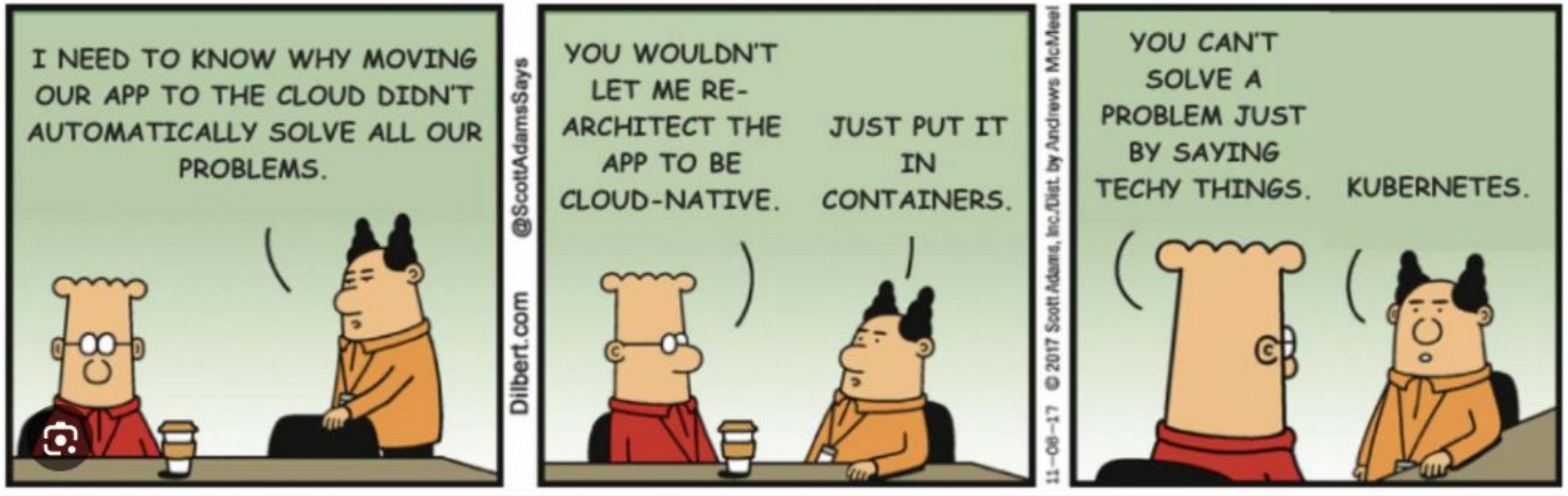


Developer Experience

Many people have been doing this on the side for years without calling it DevEx.

What about Platform Engineering?

Digital Transformations



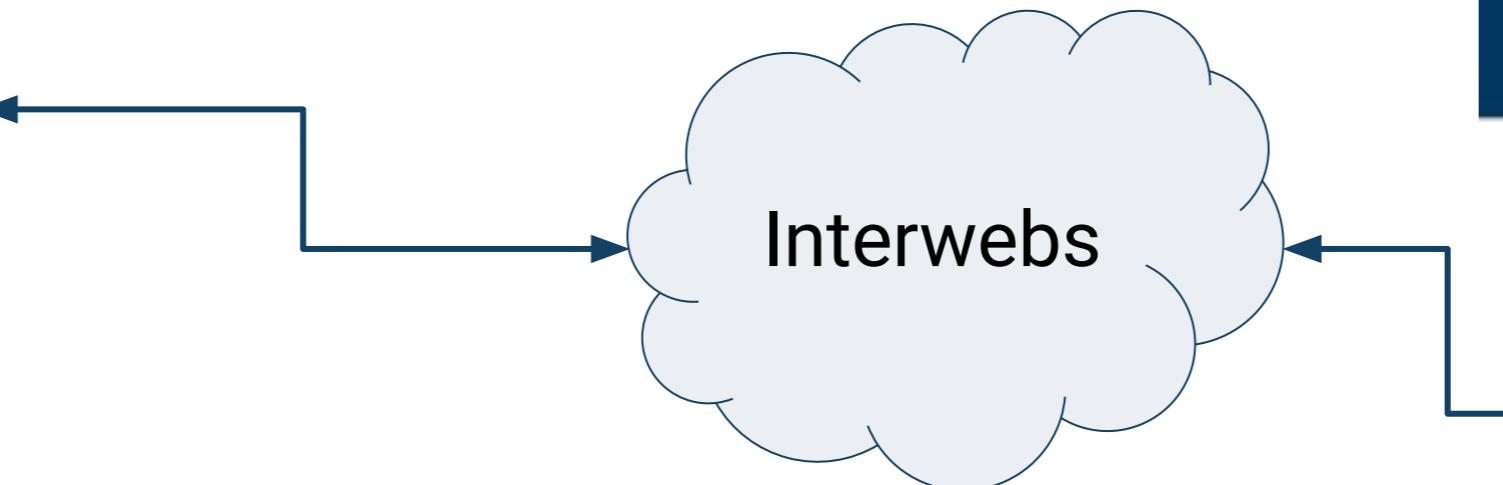
Kubernetes required to learn for first deployment

- Nodes
- Pods
- ReplicaSets
- Secrets
- Load Balancer / ClusterIP / NodePort
- Ingress Controller / Ingress Rules
- Namespaces
- Deployments
- Services
- Service Accounts and Tokens
- Service Roles and assignment

Transformations requires a mindshift

- SDLC
 - All Dev and Testing must be done in containers
 - Containerize an application code base
 - Containers are immutable - any changes require a rebuild
- Production Workloads
 - Containers can fail and restart for any number of reasons
 - Only worry about how many replicas are needed, not which “VM” it is hosted on
 - Cannot “remote in and debug” like a person could with VMs

Very easy to misconfigure something



Very easy to misconfigure something



The Chicago Way: An Electronic Voting Firm Exposes 1.8M Chicagoans



UpGuard Team

Published Aug 17, 2017

Digital Transformations

To have a successful digital transformation you
have to invest in DevEx and Platform Engineering

Avoiding Pitfalls



Platform Engineering isn't operations 2.0



Platform Engineering is not a shared service

- Shared service - developers raise tickets and wait for someone to do work.
- Processes
 - Enable developers to do work.
 - Create the processes for repeatable tasks.
 - Make it easy to spin up and create new projects.
 - Work with all parties to ensure processes are created to meet standards.
- Operation work is still required - not the purview of platform engineers
 - The application hosts are up and running.
 - Monitoring and logging.
 - Security and auditing.

Treat processes like a product
The developers and operations are the customers

Build Golden Paths not paved paths



Build Golden Paths not paved paths

- Make the process to enticing not to use it
- Involve others and come up with a plan
 - Agree there is a problem to solve
 - Identify key roles and representatives
 - Assign RACI
 - Key representatives agree on business rules
 - Make process changes
 - Pick the tooling
 - Demonstrate it
 - Roll it out

Safe and measured rollout process

- Pilot - Use the process to deploy to Production for one or two applications
- Early Adopters - Repeat for a couple more applications
 - Find flaws in the process
 - Identify shortcuts and assumptions
- General Adoption - Rollout to everyone
 - Training
 - Documentation
 - Templates

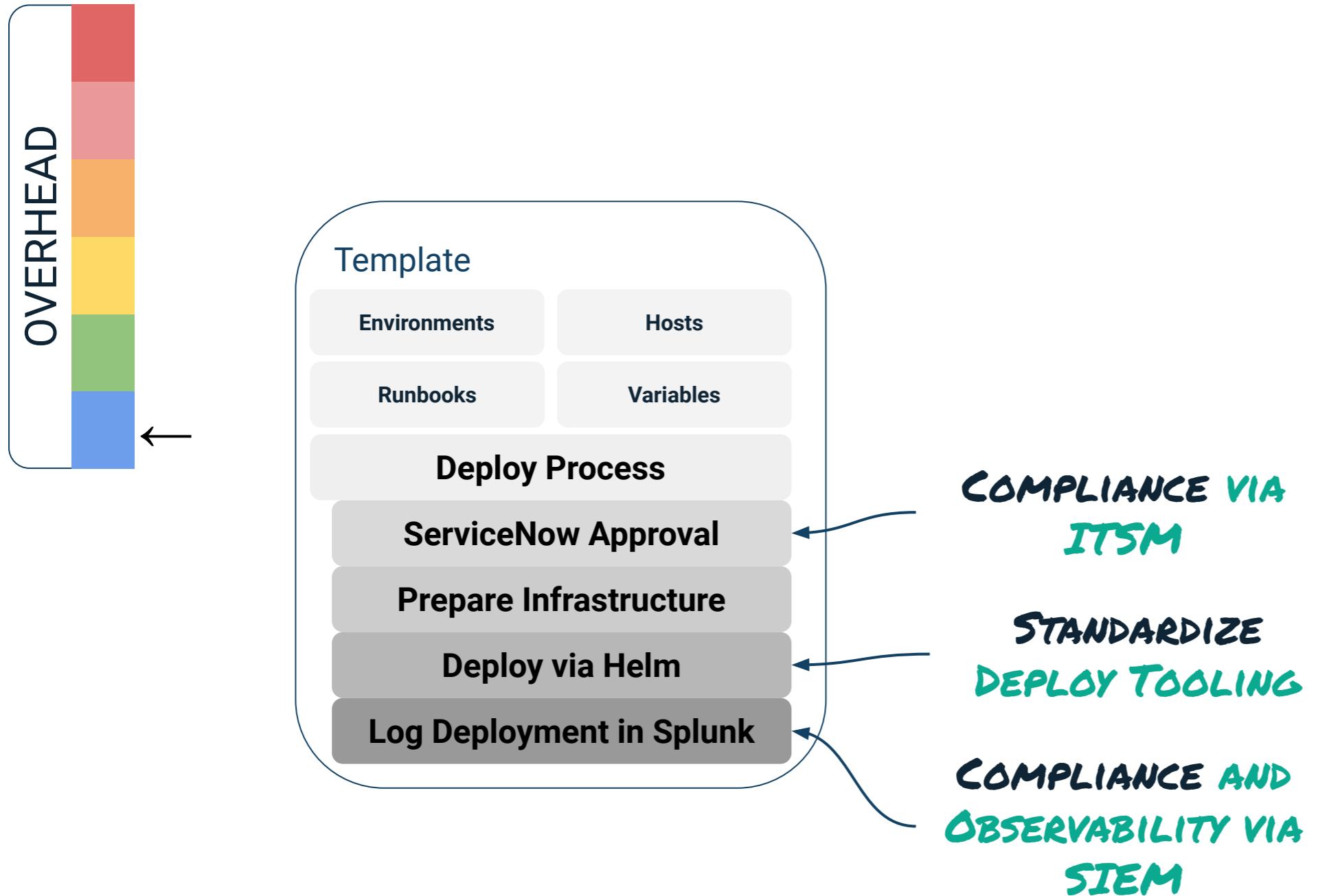
Processes should be open source
Allow developers and operations to contribute to it.

Processes are open source

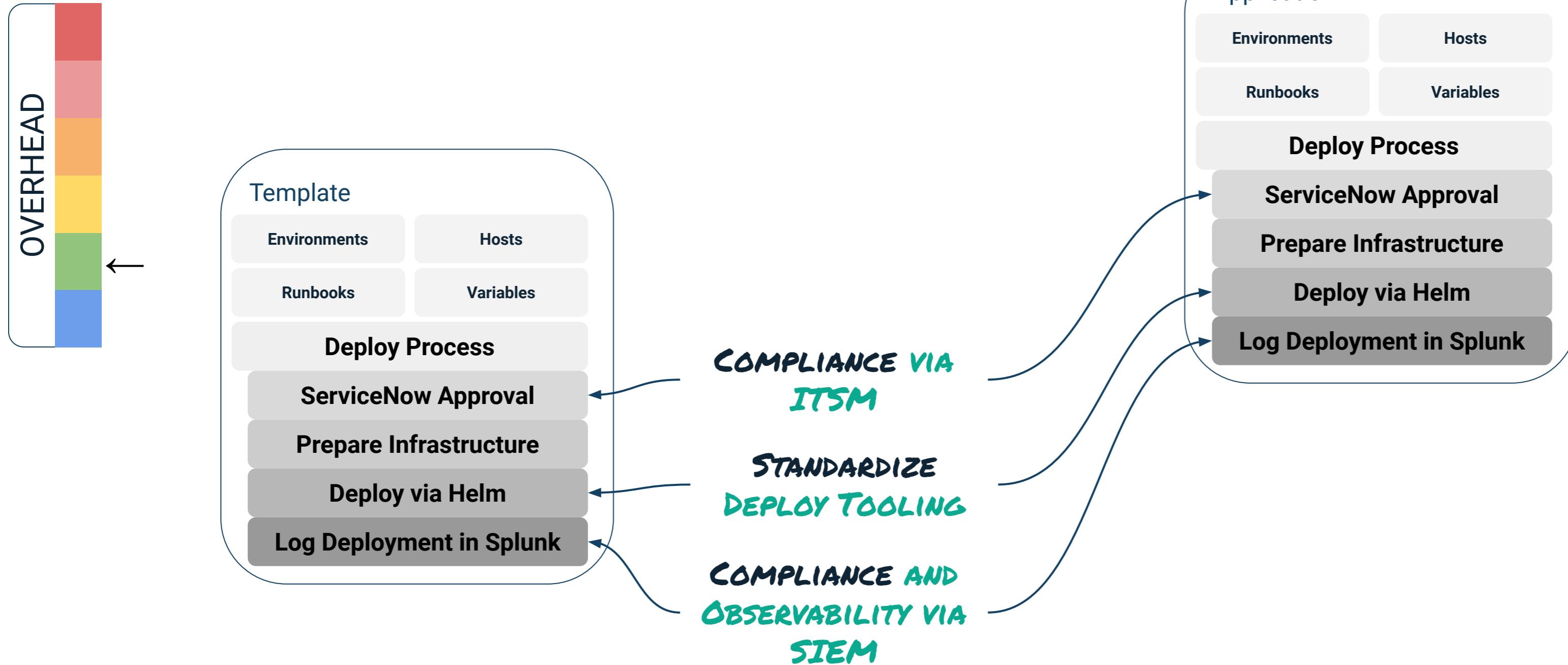
- Doesn't mean you need to use an open source product.
- Make the processes themselves open source.
 - Publish clear contribution guidelines
 - Keep documentation up to date
 - Guide contributors to solve problems with appropriate abstraction
 - Help contributors to use existing features to solve their problems

Have processes in place to rollout changes
Make changes for dev teams not to dev teams.

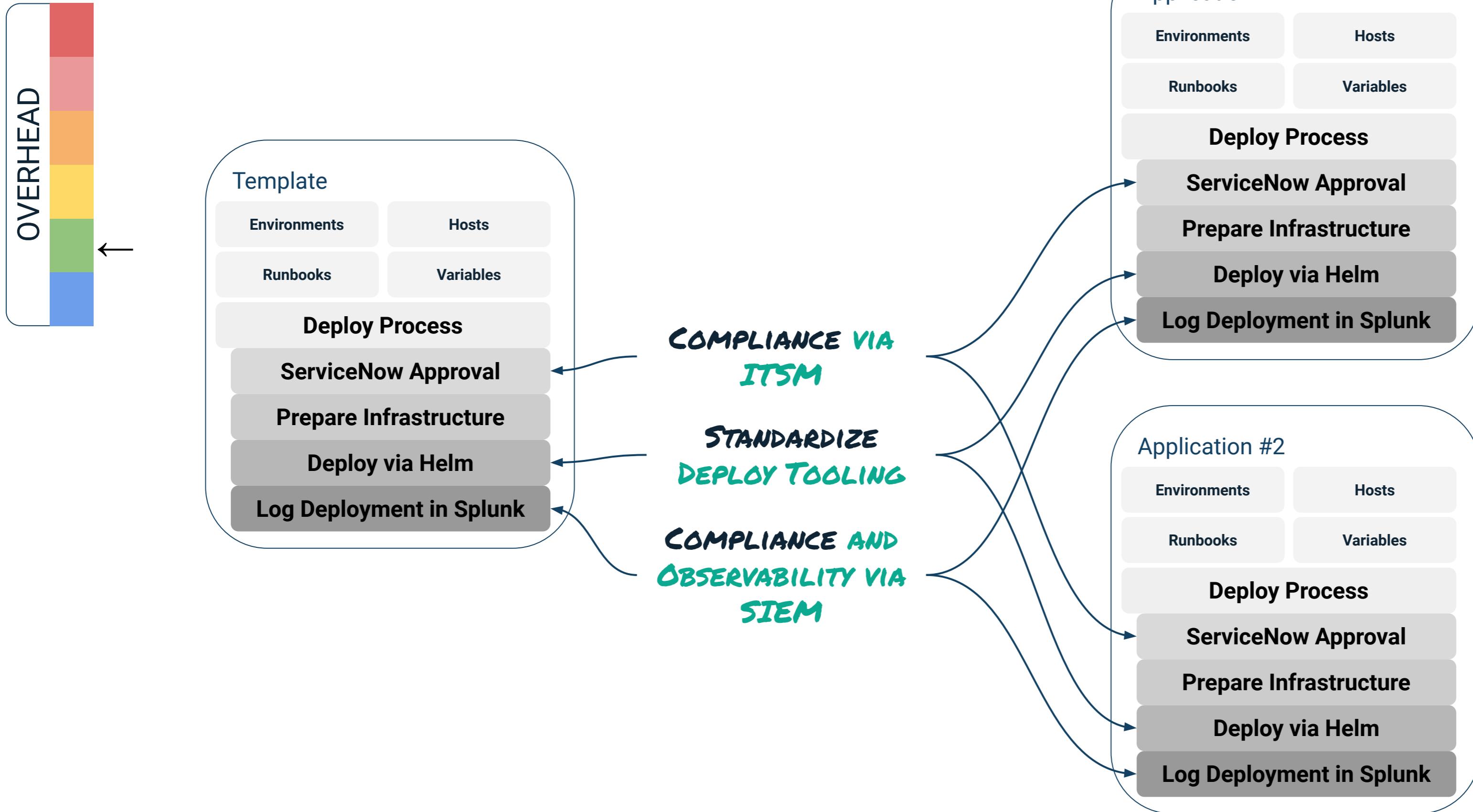
Day 0 - Platform Team creates a template



Day 1 - Dev Team clones the template for their App



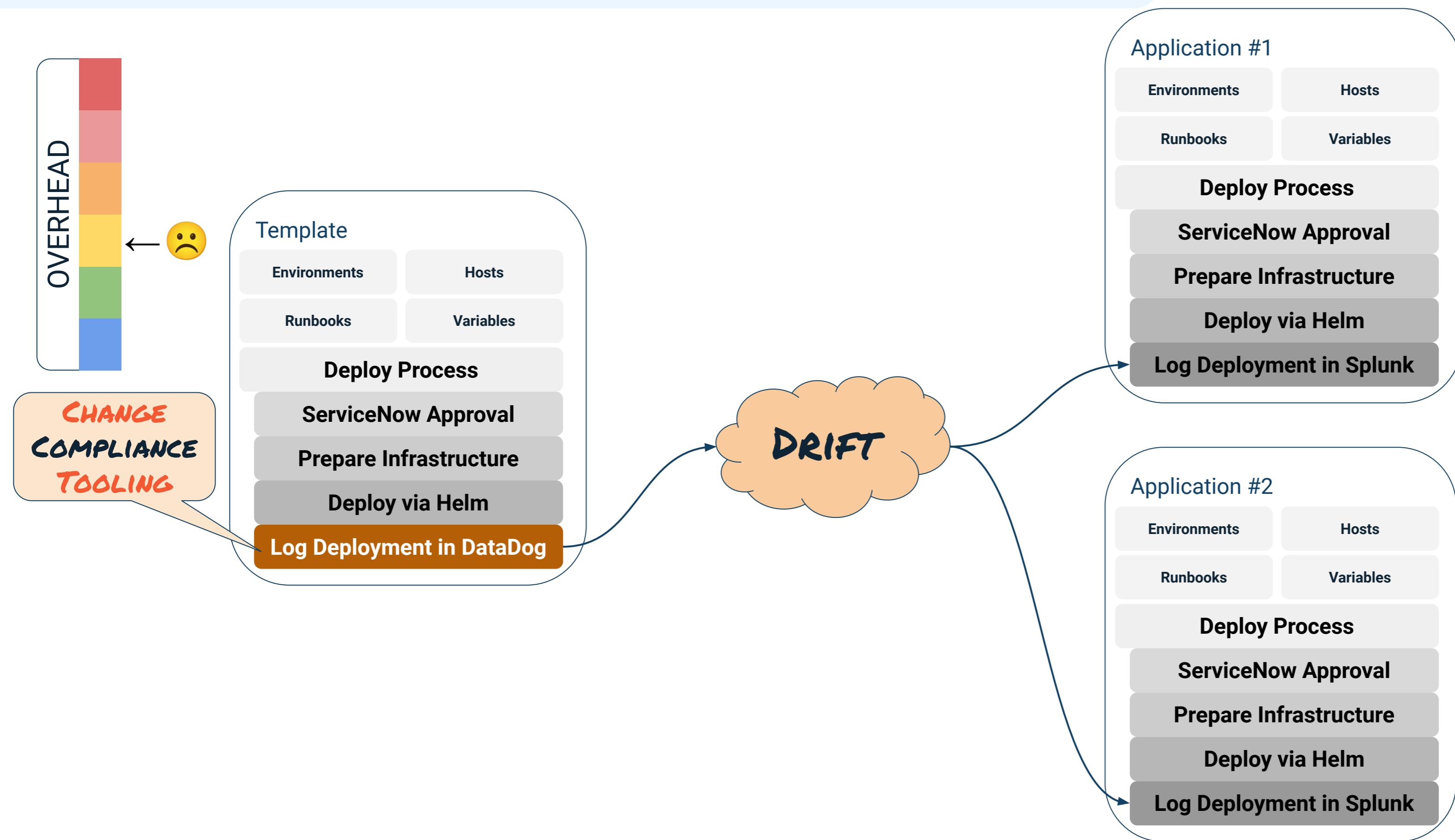
Day 2 - Another Dev Team clones the template for their App



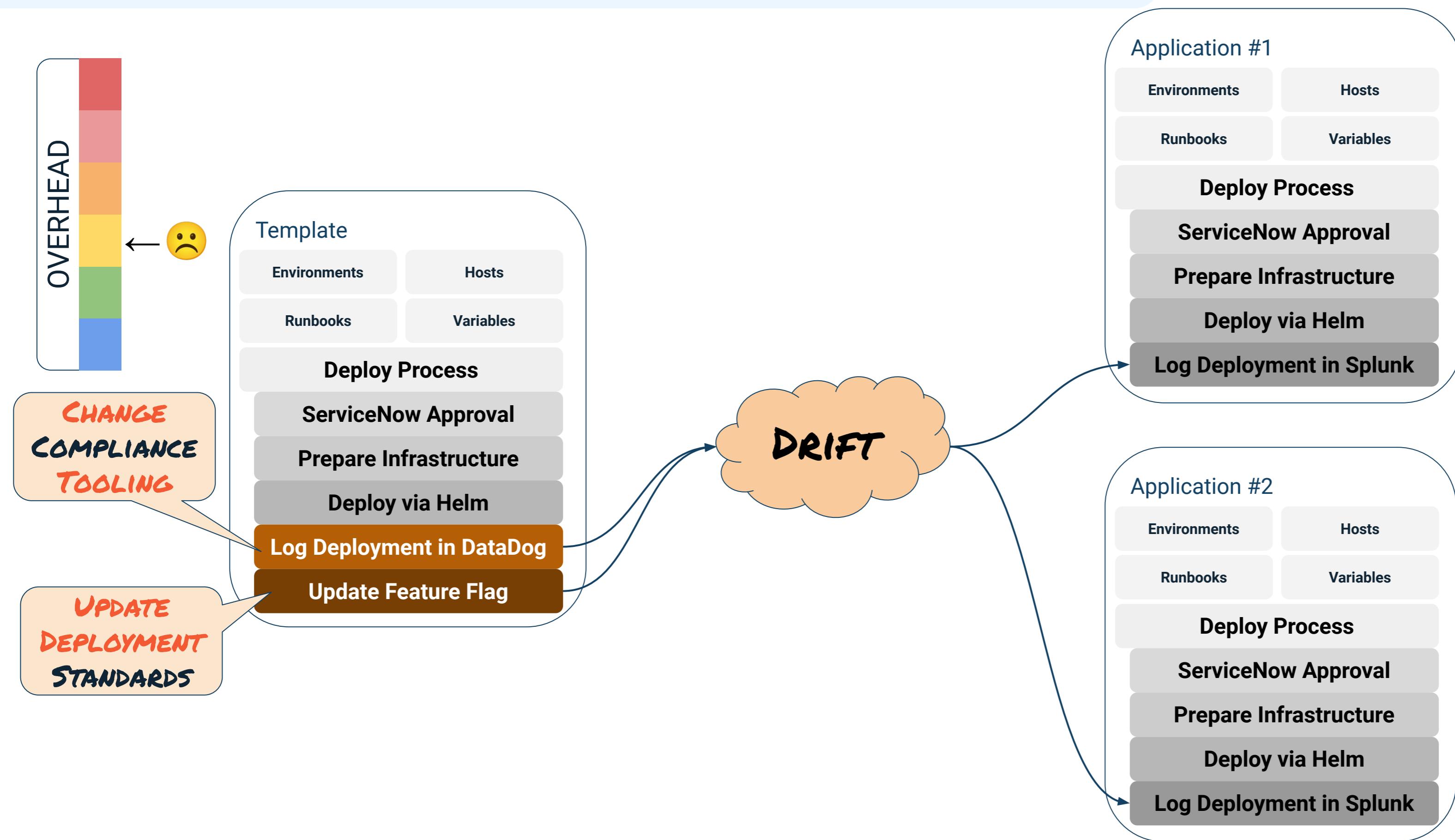
Day 30 - Platform Team Changes SIEM Tooling



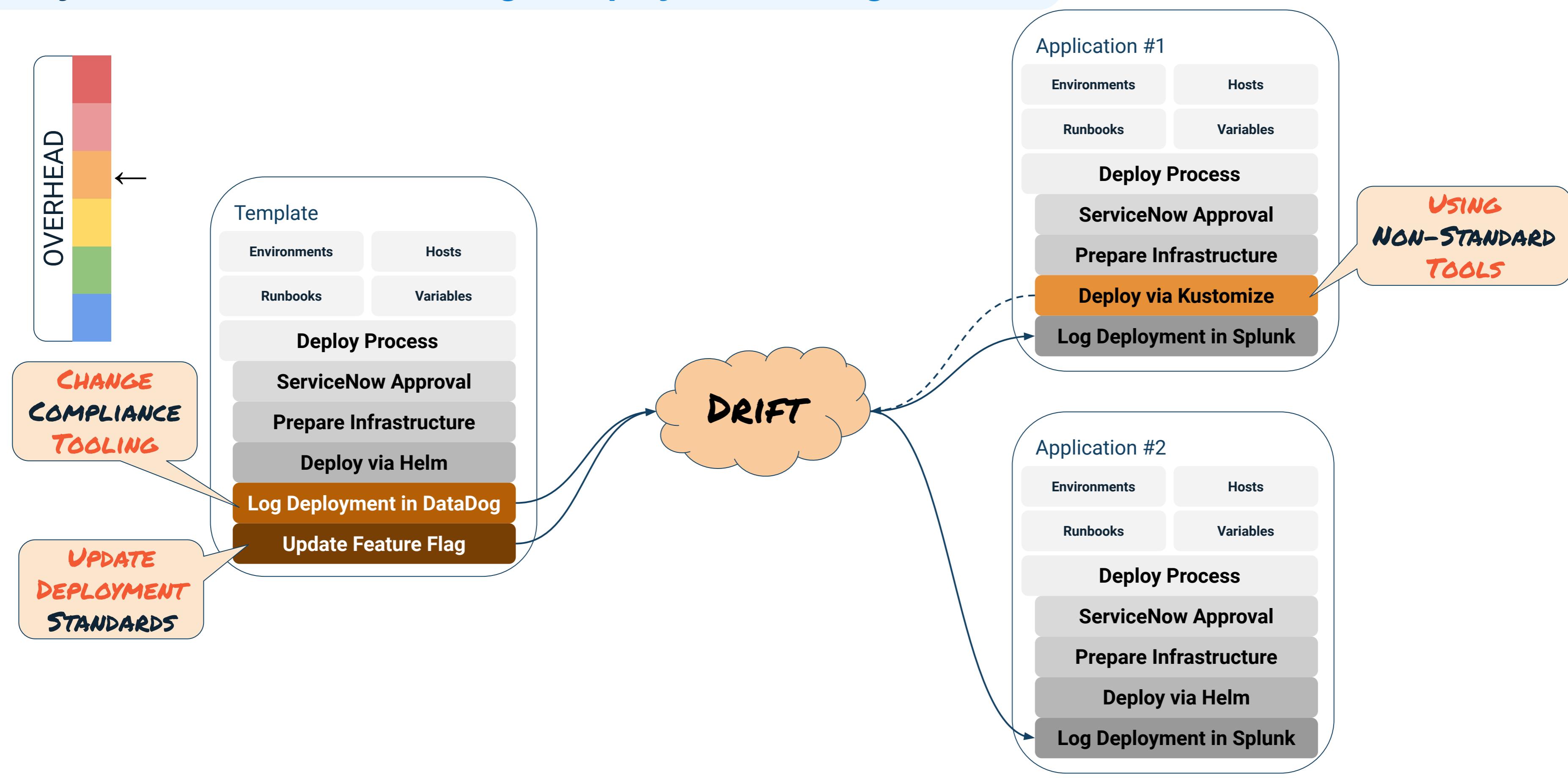
Day 30 - Platform Team Changes SIEM Tooling



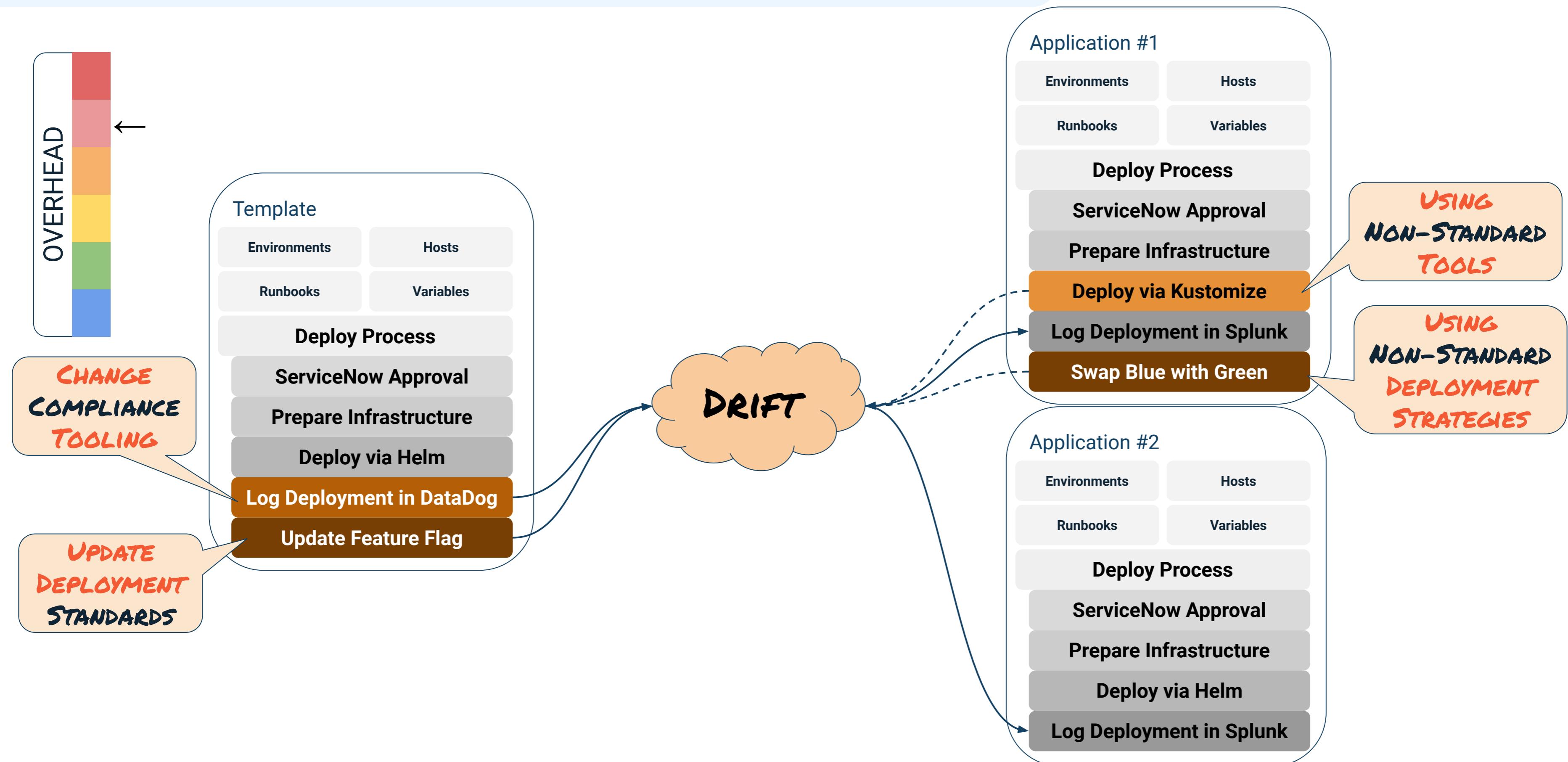
Day 50 - Platform Team updates deployment standards



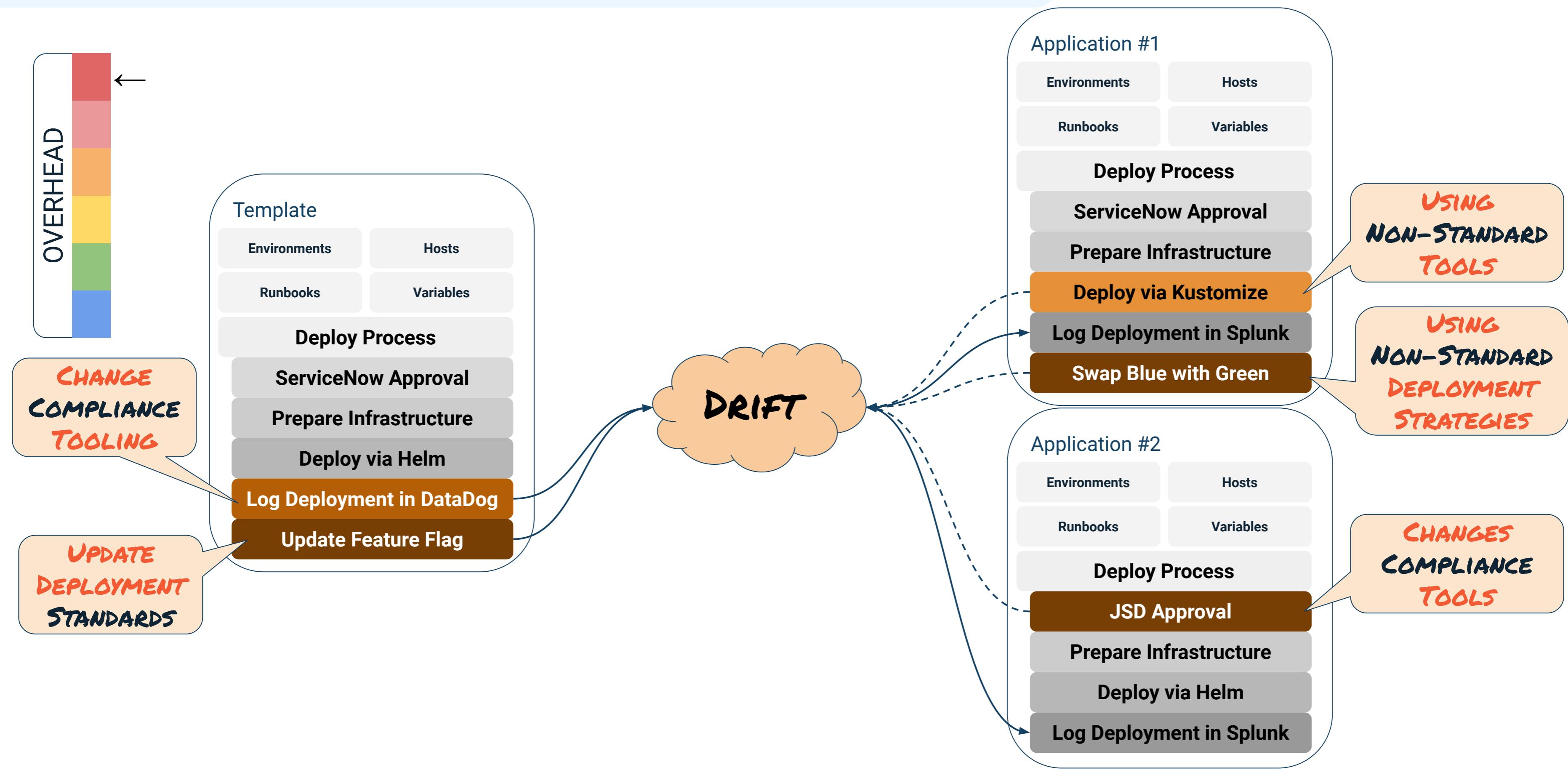
Day 90 - First Dev Team changes deployment tooling



Day 100 - First Dev Team changes deployment strategy

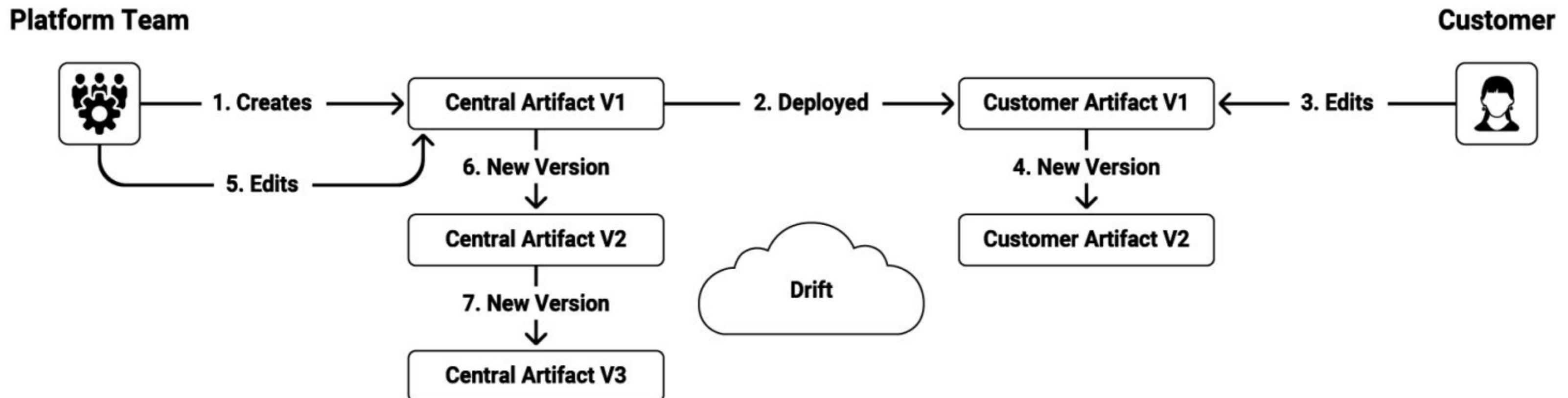


Day 120 - Second Dev Team changes ITSM tooling



Customer Responsibility Model

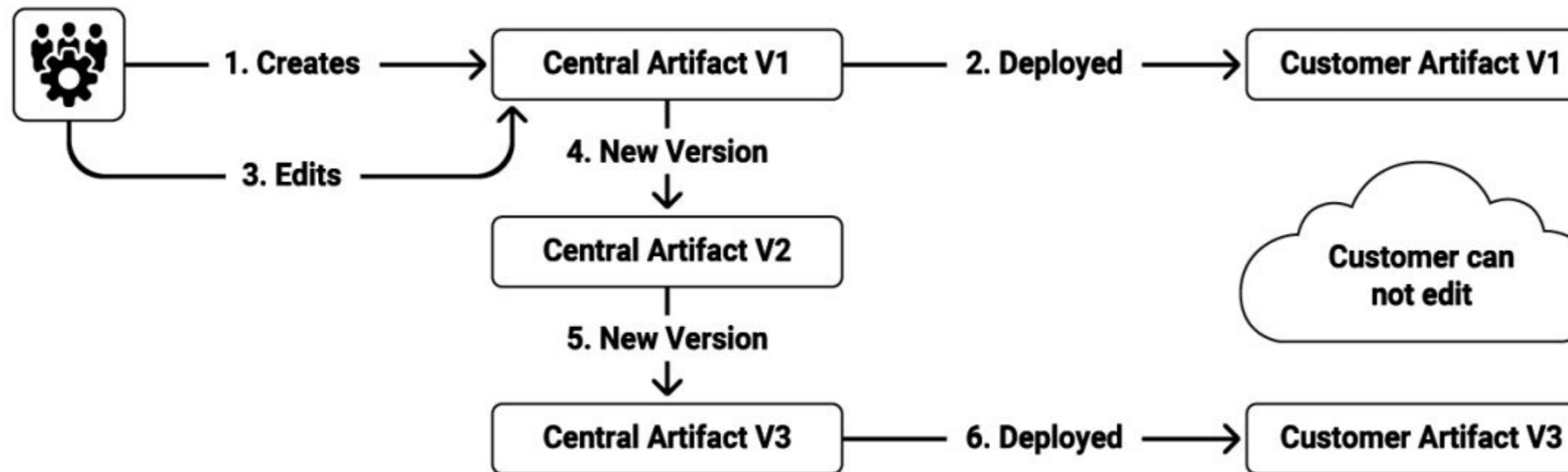
Customer Responsibility Model



Central Responsibility Model

Central Responsibility Model

Platform Team

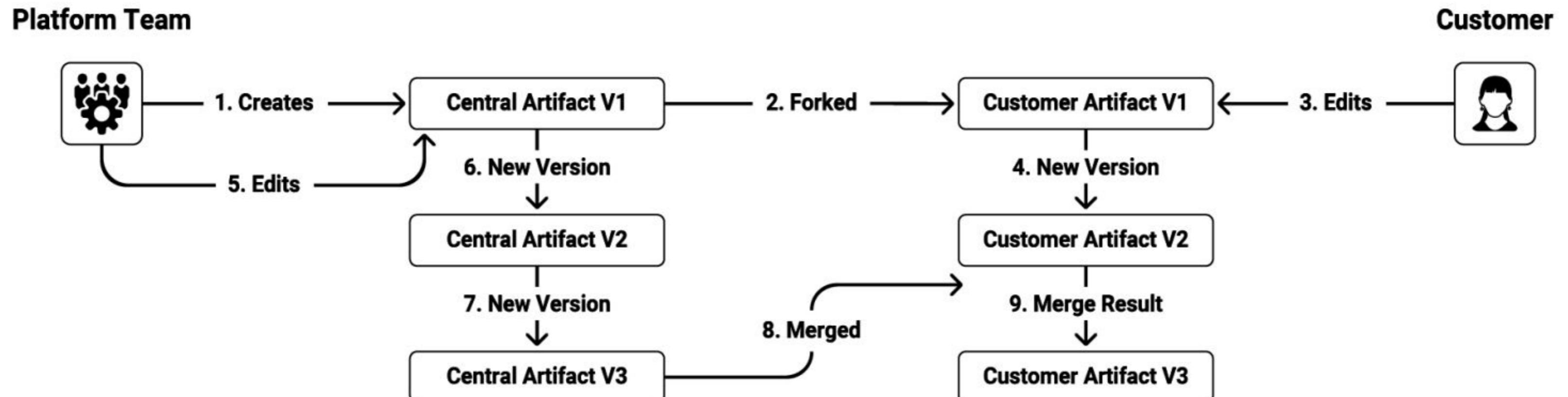


Customer

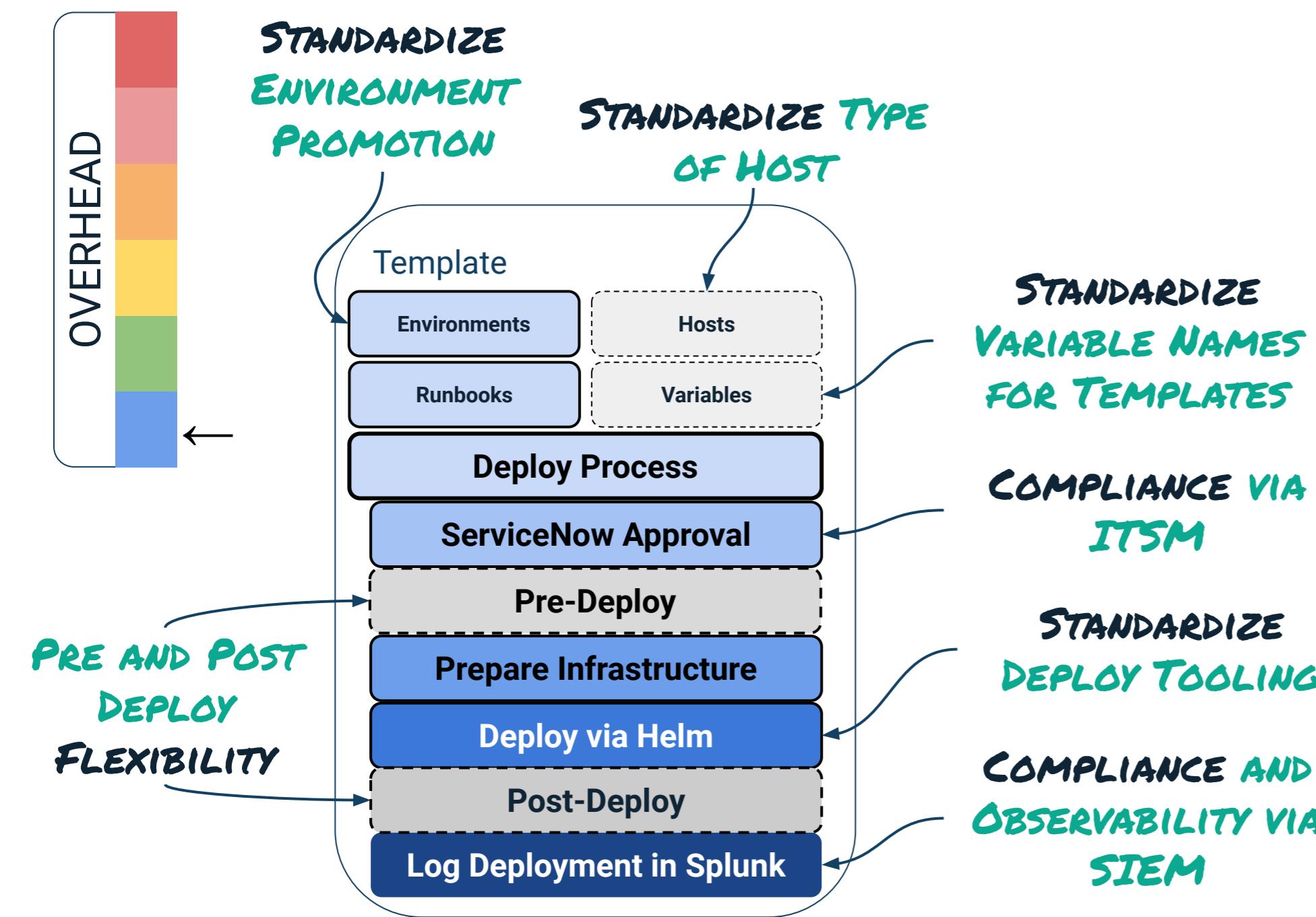


Shared Responsibility Model

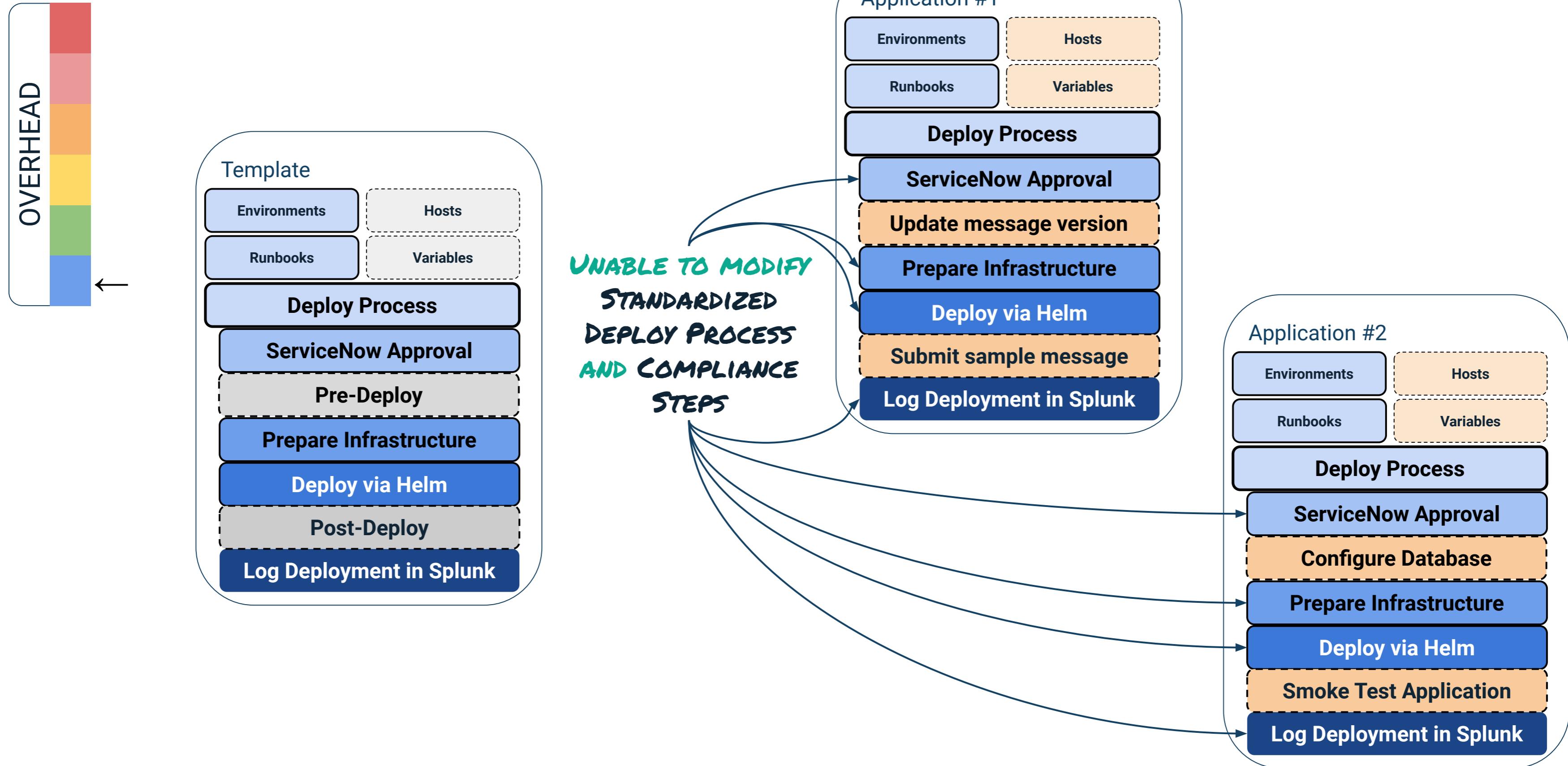
Shared Responsibility Model



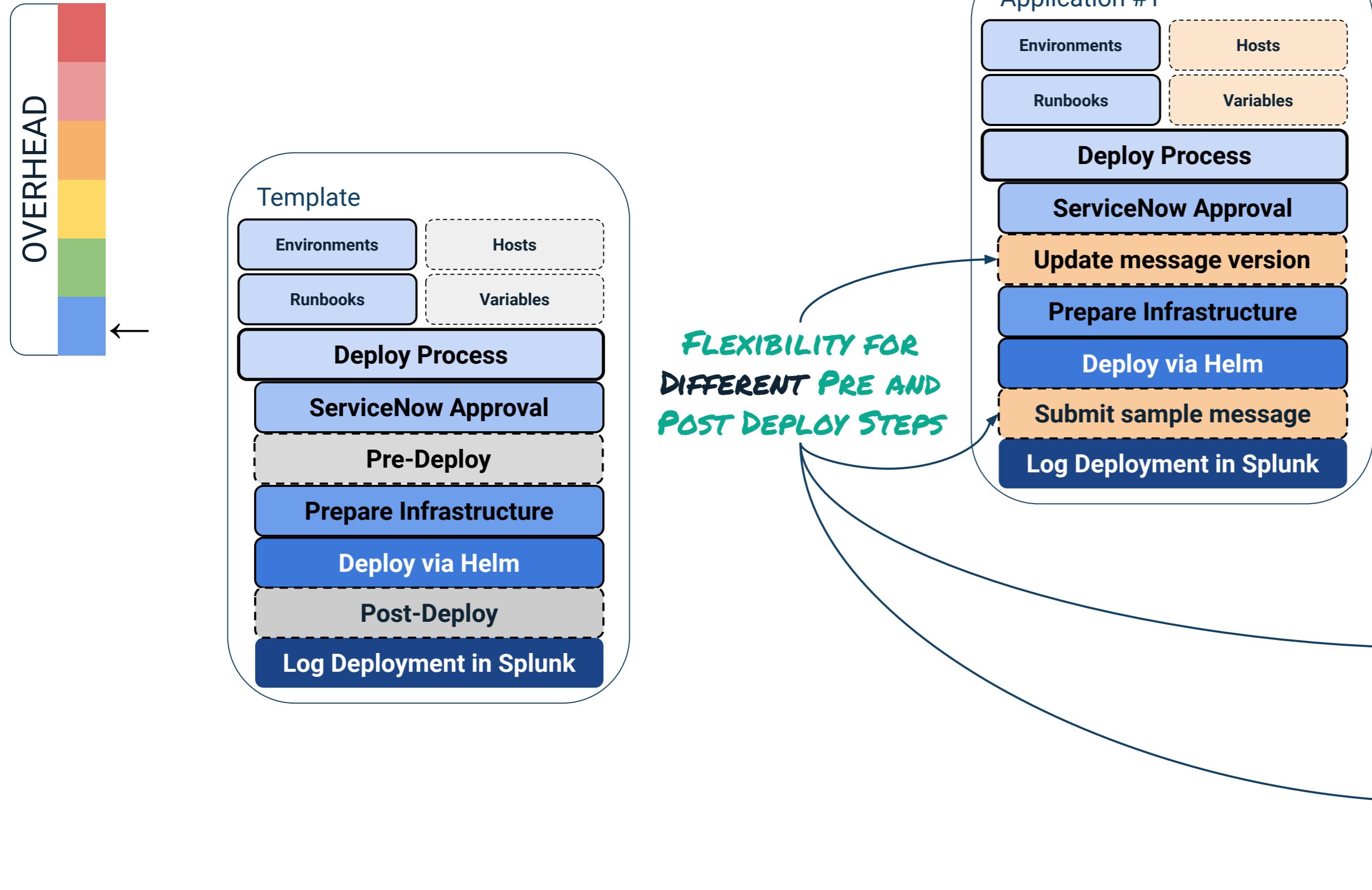
Day 0 - Platform Team creates a blueprint with guardrails



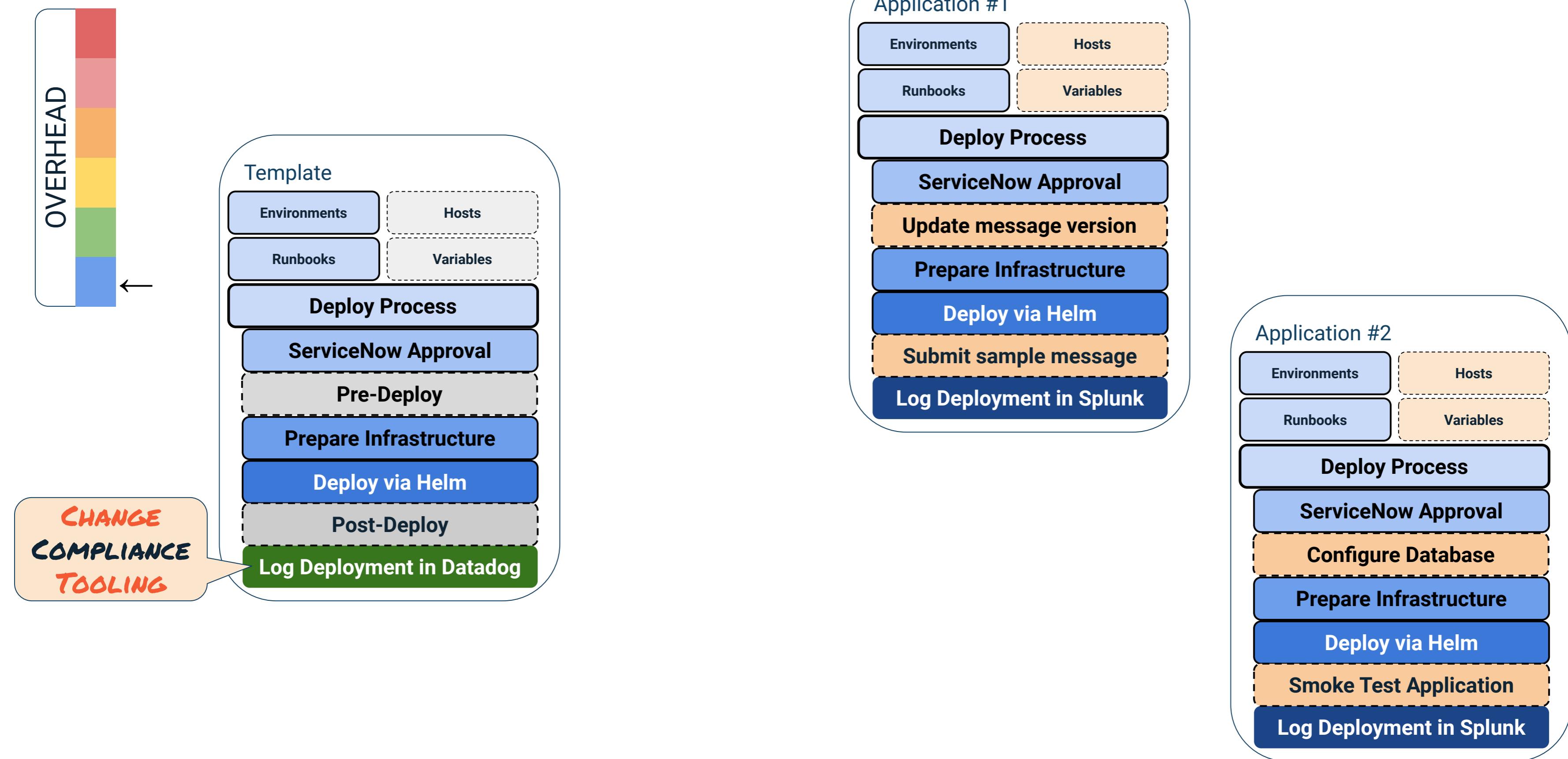
Day 1 - Dev Teams uses the blueprint with guardrails



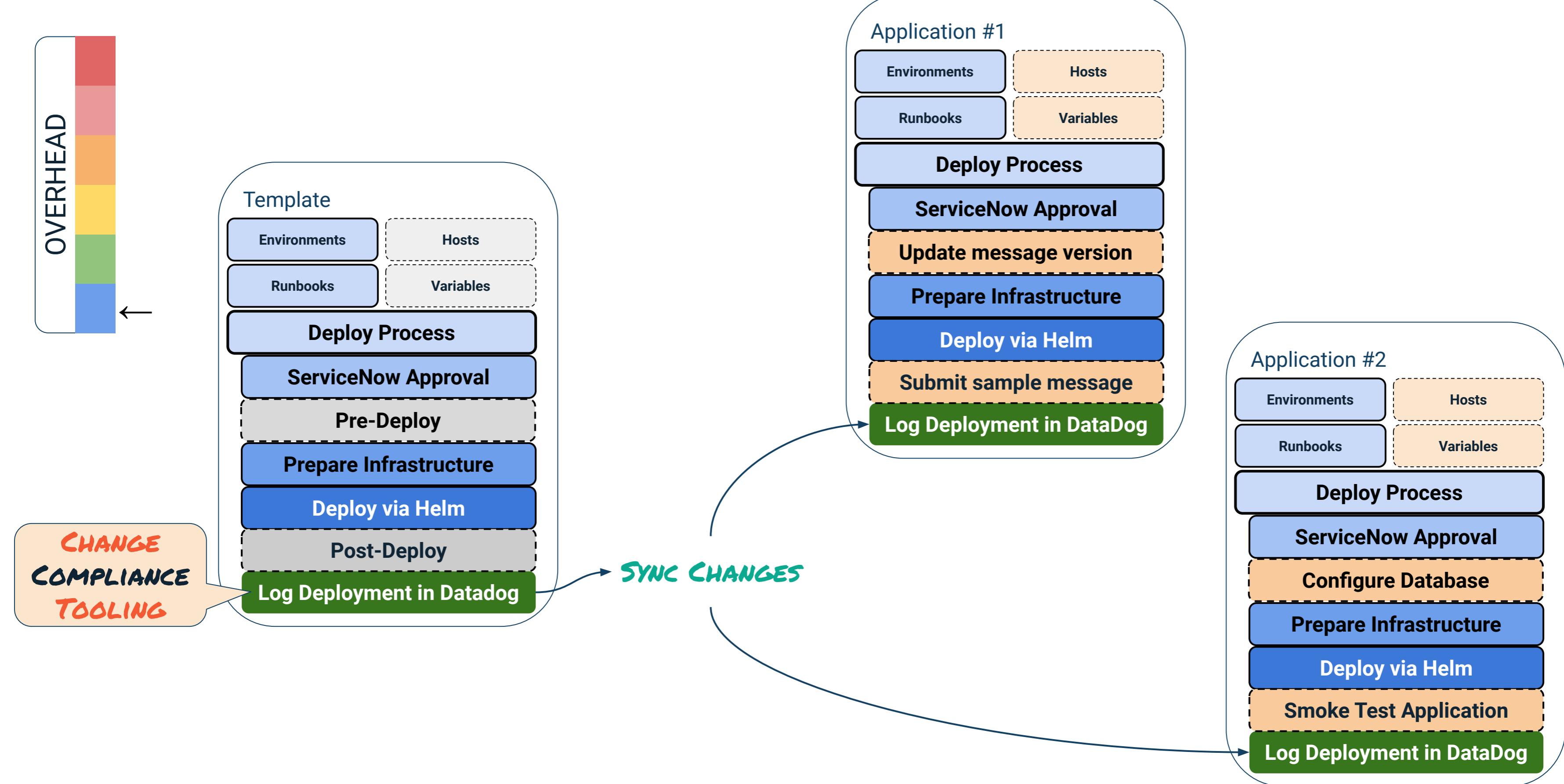
Day 1 - Dev Teams uses the blueprint with guardrails



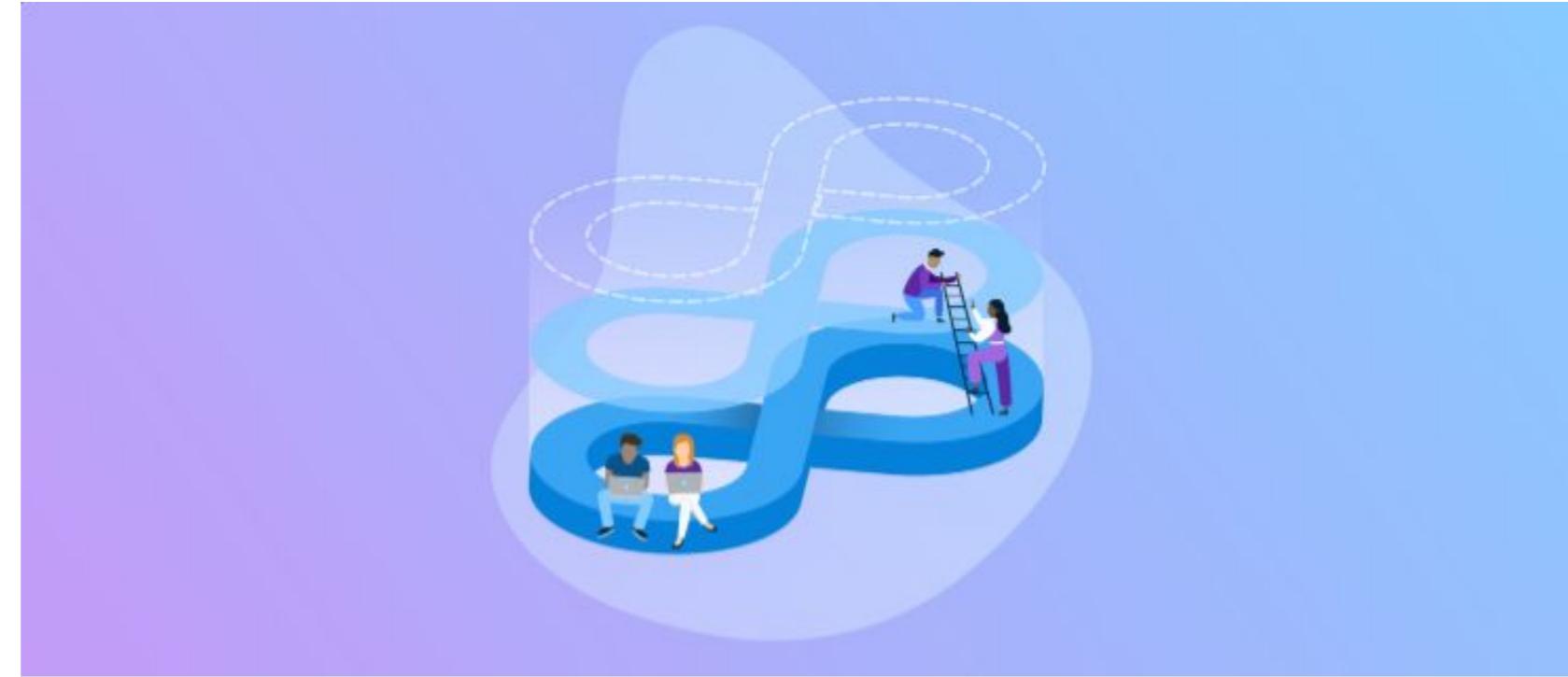
Day 30 - Platform Team updates SIEM tooling



Day 35 - Platform Team syncs updates to child apps



Not an easy problem to solve



Blueprints to empower Platform Engineers



SHIPS MAHINDRA | Posted on May 2

Submit your ideas on how Blueprints could benefit you below!

Over the past few months we have been learning about the needs of Central DevOps or Platform teams that own and maintain Octopus. Based on that, we heard that Platform teams want to:

- Reduce the onboarding or getting started time for a new pipeline
- Have a consistent pipeline that is easier to maintain

We are investigating Blueprints for Process and Runbooks to enable standardising on a grouping of Steps that can be reused across multiple projects. Related ideas that we are exploring are:

- How to validate Processes by defining rules / guardrails
- How to roll out changes to these Blueprints
- How to view usages of Blueprints and impact on team metrics



Processes should be easy to use
The higher the friction, the slower the adoption.

Can't solve company culture problems
No tool on earth can fix “I don't trust developers.”

Measuring Success



You have to measure to know the impact



Tangible Impact

- New people
 - The F5 Contract
 - Time to first commit
- Feature work
 - Lead time for changes
 - Deployment frequency
 - Change failure rate
 - Mean time to recovery
- Engineering NPS

DORA Metrics are a good start

But sometimes you want to go a bit deeper

Measuring Success of feature work at Octopus

- PR build duration
- Lead time to changes
- Batch Sizes
- Mean time to recovery (fix a defect)
- High severity defect escape rate (commits causing high severity issues)

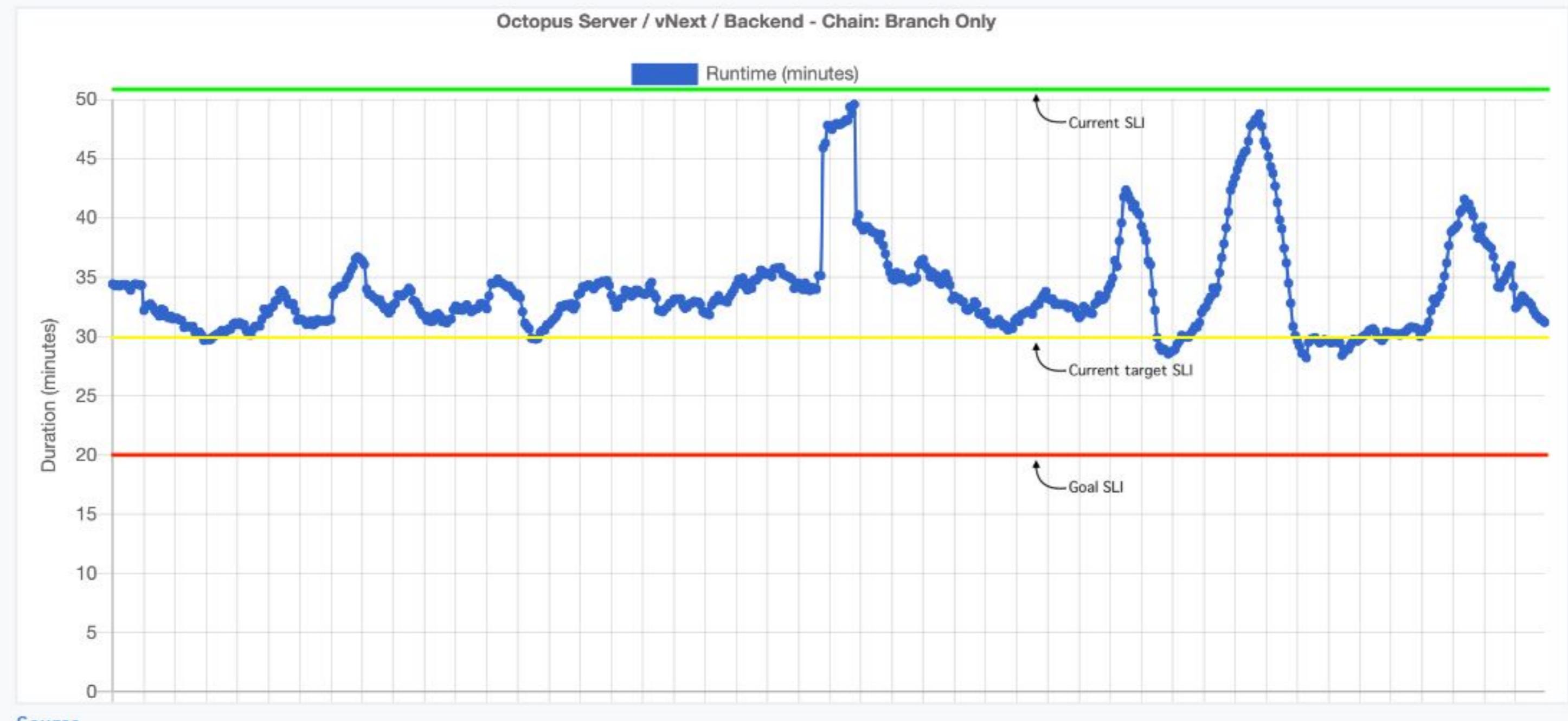
PR Build Duration

PR Build Duration

● Current SLI: < 75 minutes [[SLI details](#)]

● Current target SLI: 30 minutes [[The 30 Minute Challenge](#)]

● Long term goal: < 20 minutes

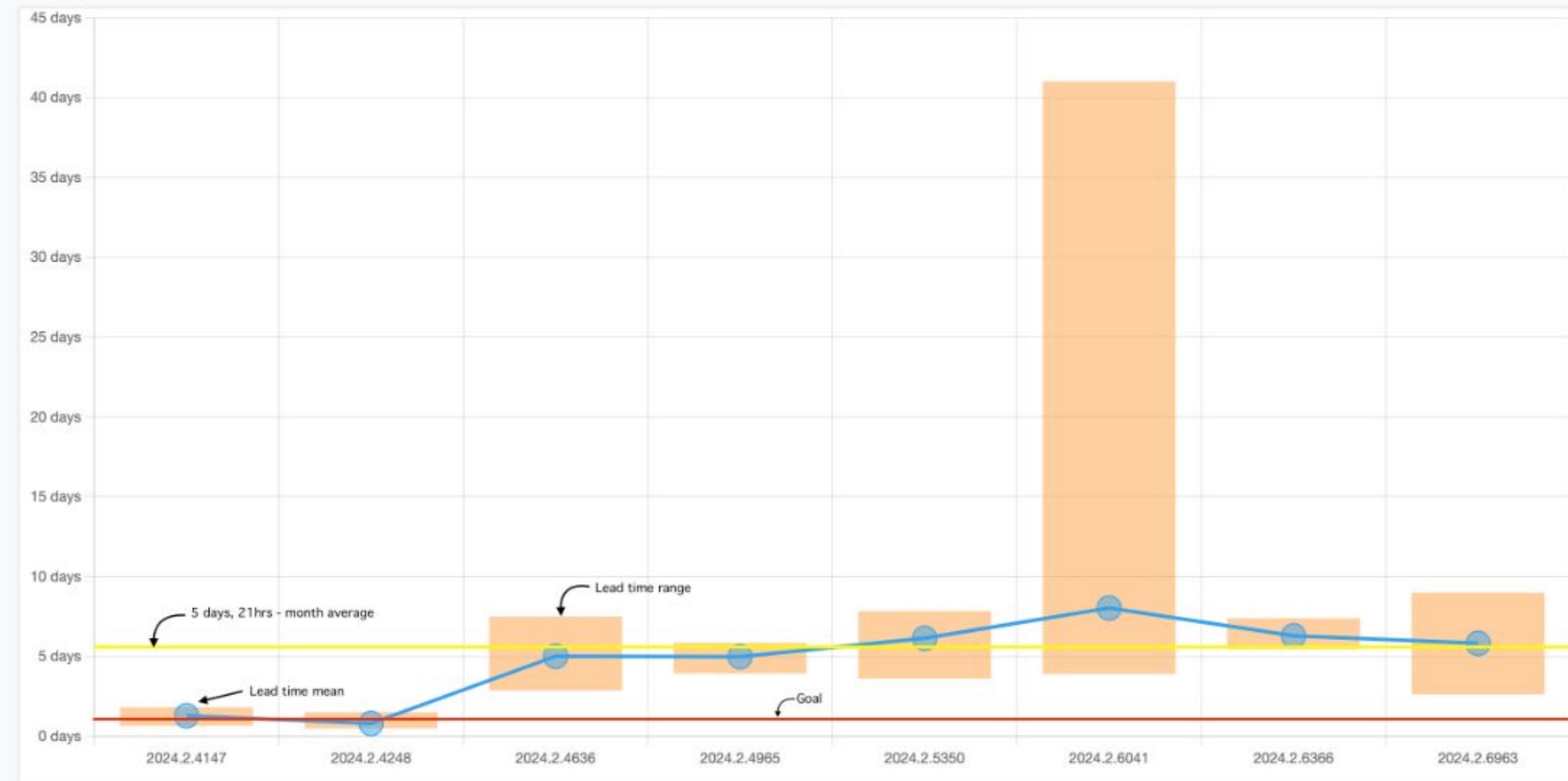


Lead Time to Changes

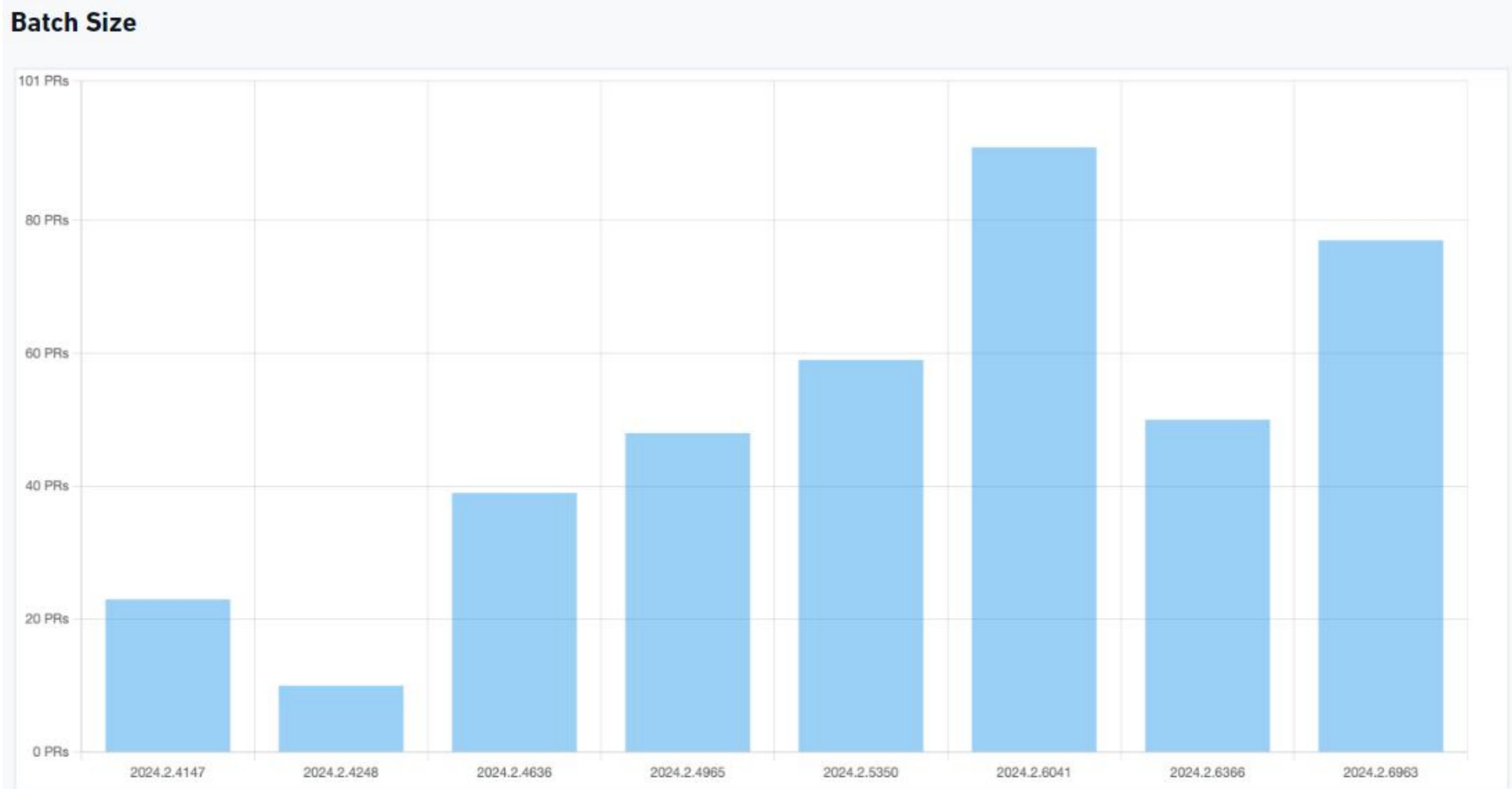
Lead time for changes

● Current measure: Month average 5 days, 21hrs. [Details]

● Goal: 1 day



Batch Sizes



Mean time to recovery (MTTR)

Current measure [Details]:

- 4 days average to detect
- Less than 1 days to fix, with outliers being over weekend or holiday period

Goal: < 1 hour to fix



Source

Ignore the -5 days for now, looks to be a bug in the graphing code.

#6542 had a long time to detect as the bug was related to data in customers database that wasn't expected and when changes as part of the EF migration.

**Platform Engineers are responsible for
gathering and visualizing this data**

Resources



Resources

- <https://DeveloperExperience.io>
- <https://dora.dev>
- <https://octopus.com/devops/platform-engineering/>
- <https://octopus.com/blog/paved-versus-golden-paths-platform-engineering>
- <https://octopus.com/blog/devops-platform-engineering>
- <https://learn.microsoft.com/en-us/platform-engineering/what-is-platform-engineering>
- <https://microsoft.github.io/code-with-engineering-playbook/developer-experience/>
- <https://github.blog/2023-06-08-developer-experience-what-is-it-and-why-should-you-care/>



 @bobjwalker

 bob.walker@octopus.com

 @bobjwalker

 bobjwalker.octopus.app



Thank you!

Bob Walker - Field CTO at Octopus Deploy

@bobjwalker

bob.walker@octopus.com

@bobjwalker

bobjwalker.octopus.app

