

# DevOptimise

OPTIMISE THE WORKFLOW OF DEVOPS PROJECT



Yash

BACKEND DEV



Sean

BACKEND DEV



Michael

FRONTEND  
DEV



Harry

FRONTEND  
DEV



Bianca

RESEARCH



Lino

RESEARCH

# Agenda

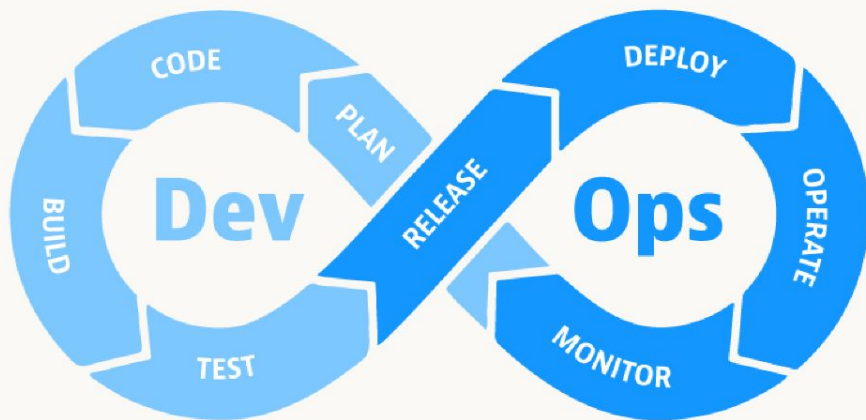
- 
- What is DevOps?
  - Problem Statement
  - Real-time Demo
  - Solution Architecture
  - Roadmap
  - Lessons Learnt
  - Questions



## Meet Larry

- Larry is overseeing 20 projects using a DevOps framework
- Wants to minimise time spent on resolving merge conflicts and in PR reviews
- Wants to check if developers are following the merge strategy
- Doesn't have time to host standup meetings to go through each of the projects in depth

# What is DevOps?



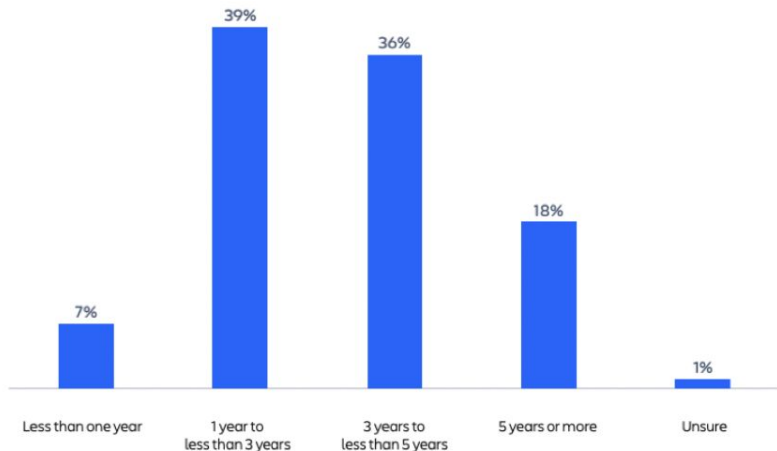
Aims to achieve:

- Improved collaboration
- Rapid deployment
- Improved quality and reliability

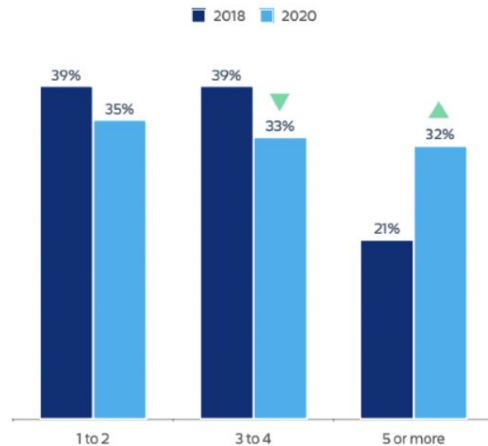
# The DevOps Market

- DevOps stakeholders are growing [2021 Upskilling Enterprise DevOps Skills Report]
- DevOps Toolchains on market are complex and not effective enough

**LONGEVITY OF DEVOPS PRACTICES**



**NUMBER OF TOOLS CONSULTED TO UNDERSTAND PROJECT STATUS**



# DevOptimise Problem Statement

*How might we gain insights to help  
optimise the workflow of DevOps projects?*

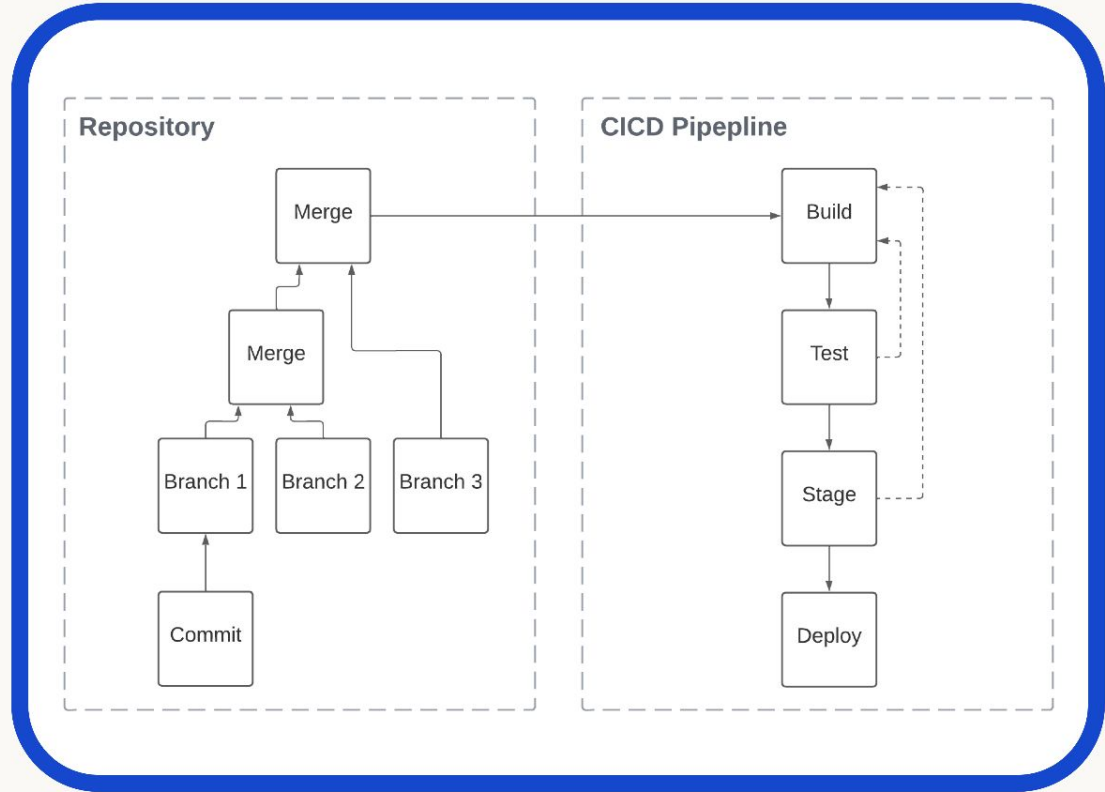
# The DevOptimise Solution

Improved repository health with analysis of key metrics

Improved reproducibility by comparing pipelines

Scalability: can quickly gain insights to the workflow and see what is working across large or multiple teams

# Key Areas Targeted



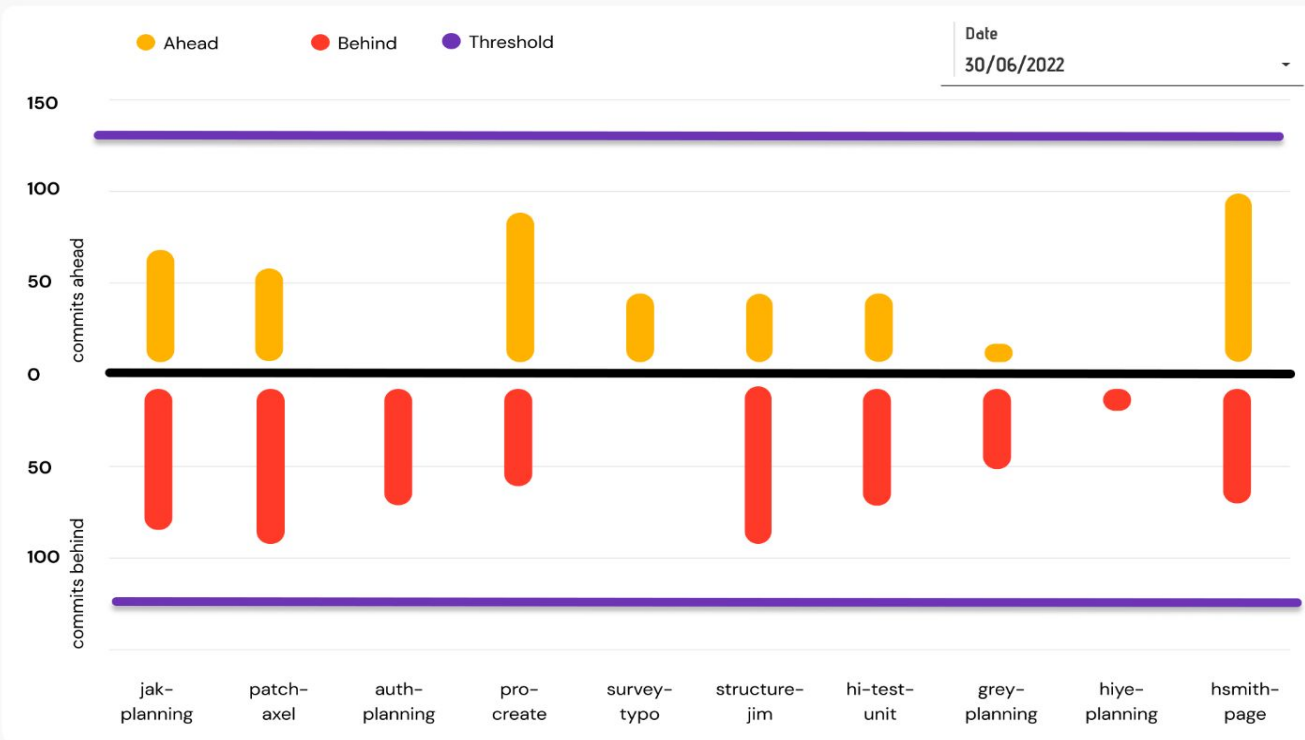




# DevOptimise

## Repo Health

### Commit Behind/Ahead Tracking



### Health Metrics

3.8 days

Average Merge Delay

0

Branches Violating Threshold

1.8 days

Average Pull Request Review

7.5 days

Average Branch to Main Merge Time

Source:

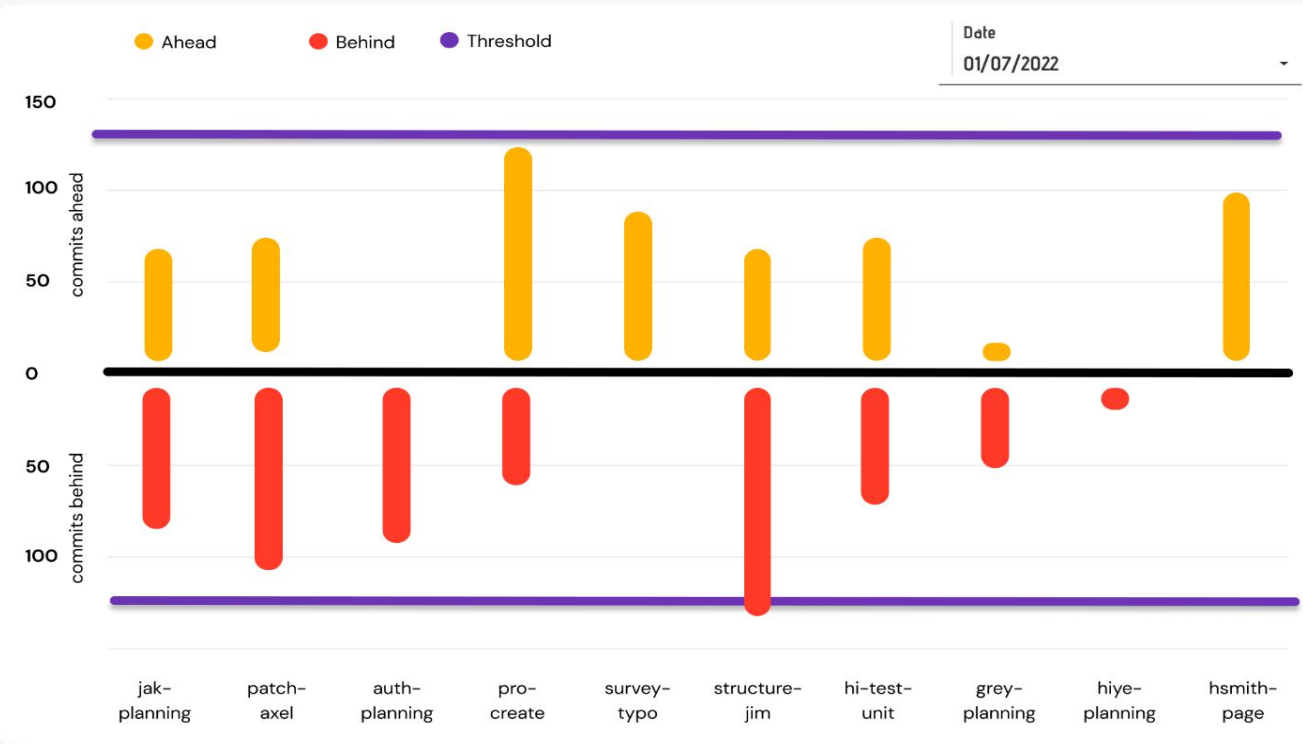




# DevOptimise

## Repo Health

### Commit Behind/Ahead Tracking



### Health Metrics

3.8 days

Average Merge Delay

1

Branches Violating Threshold

1.7 days

Average Pull Request Review

7.5 days

Average Branch to Main Merge Time

Source:





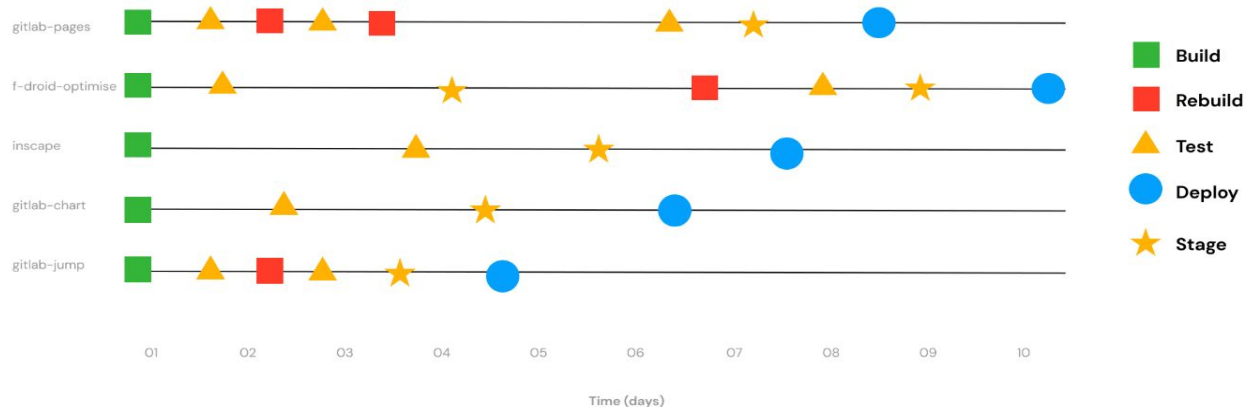
# DevOptimise

## Pipeline Track, Trace & Compare



Date

01/07/2022



36

Jobs in Pipeline

1.1 days

Average Job Length

3.5 days

Longest Job

Rebuild

Phase with most Errors



# DevOptimise

## App Release Traceback



git-lab-chart-heatmap-3.2  
24/05/2022

User Acceptance Testing  
20/05/2022

QA Pass  
18/05/2022

Test  
10/05/2022

ut-feat-bi-vis\_09/05/2022\_john\_doe

ut-feat-line-plot\_09/05/2022\_john\_doe

ut-feat-overlay\_08/05/2022\_jain\_doe

Build  
06/05/2022

job-compile-feat-ban\_04/05/2022\_yash\_patil

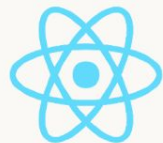
job-auto-adjust-heat\_04/05/2022\_sean\_chen

job-auto-style\_02/05/2022\_harry\_blackburn

*[DEMO]*

# Solution Architecture

Frontend



React



Backend



Amazon Athena



Amazon S3



Data  
Source



# How is our product different?

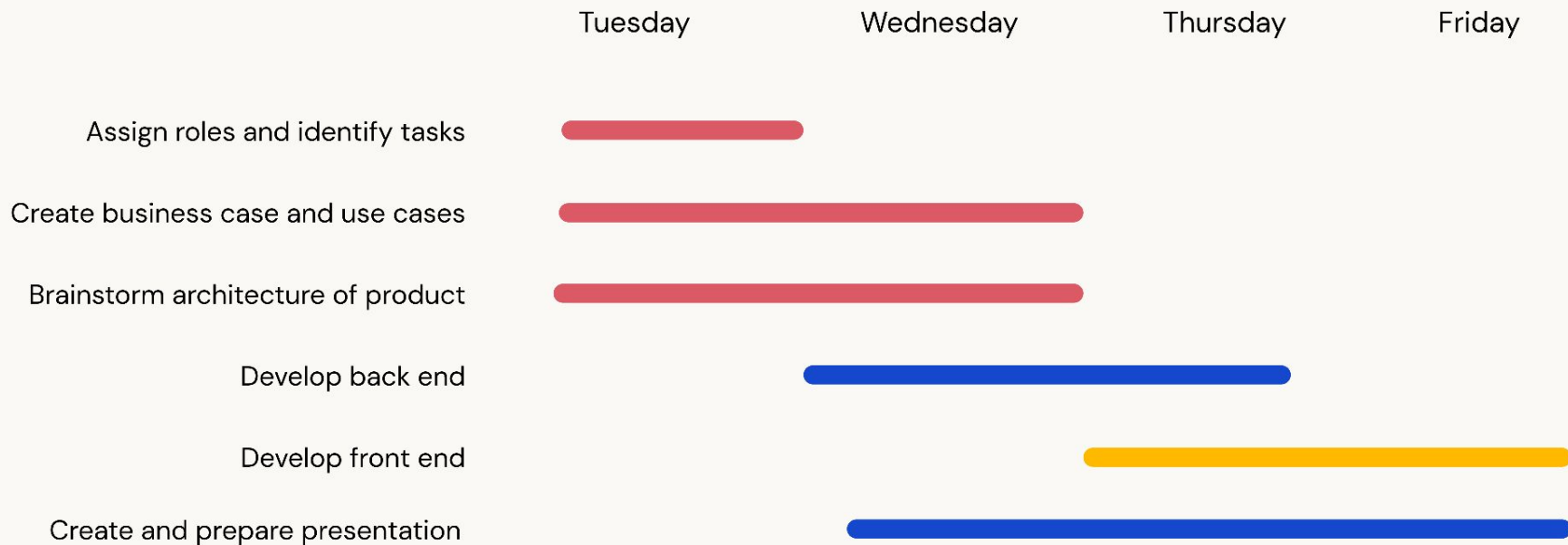
- Metrics available show what is happening but not why they're happening
- No information about merge delay time or any branch tracking capability
- No CI/CD pipeline visualization capabilities
- No breakdown of CI/CD metrics
- No simple way to determine the best pipeline format for a particular project

# Future Developments

- 1 Machine learning algorithms to determine the most optimal pipeline format
- 2 Push notifications to manager's phone when the repo health is poor
- 3 More indicators of repo health beyond commits behind/ahead



# Team Process



# Lessons Learnt

- Effective collaboration within a team with both technical and non-technical backgrounds requires extensive communication
- Agile project management method is effective for developing software projects
- New tools and technologies don't always work

Q + A