



# Agile Data Engineering

## Reliability and Continuous Delivery at Scale



**Richa Singhal**

Senior Data

Engineer

Go-To-Market Data Engineering

**ATLASSIAN**



**Esha Shah**

Data Architect

# Agenda

**Our Journey and scaling  
pains**

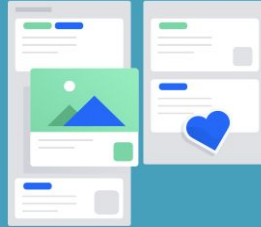
**Adapting to scale reliably**

**Wins , current challenges  
and takeaways**

 Confluence



 Trello



 Bitbucket



 Opsgenie



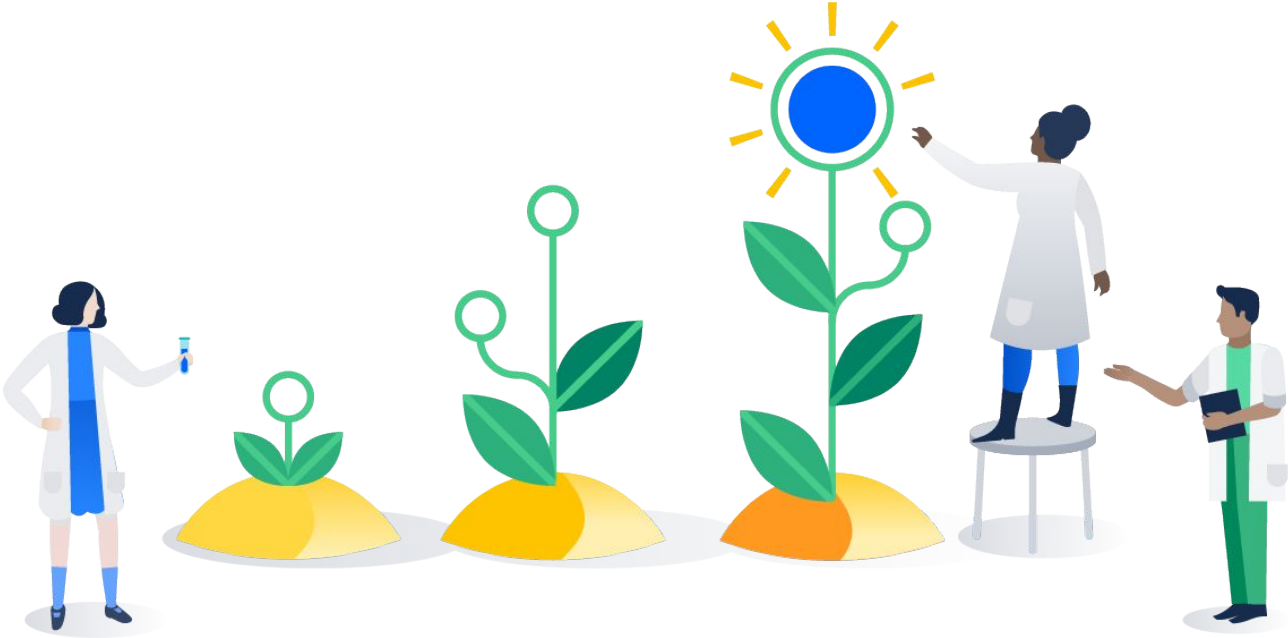
 Statuspage



 Jira



# Our Journey



# Last Decade

Postgres Datawarehouse

(Shell scripts, SQL)

2012 - 2013

Single Data Lake

(S3, EMR, EC2)

2016 - 2017

Data lakehouse

(Delta, Fivetran\*, Unity catalog\*)

2020 - Current

2014 - 2015

Postgres Datawarehouse

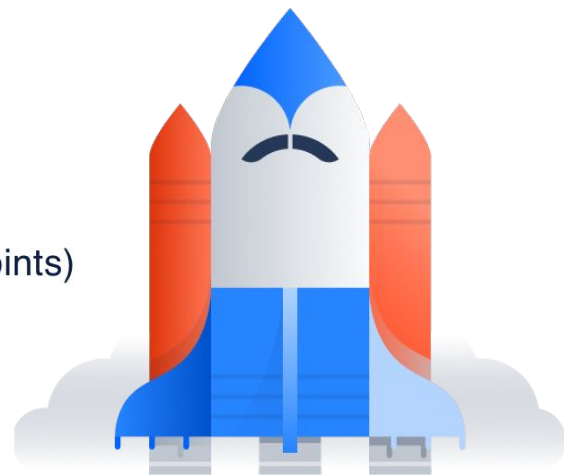
(Pentaho Kettle)

Redshift event store

2018 - 2019

Single Data Lake

(Databricks, SQL endpoints)



# Scaling Pains

---



**Unreliable data**

**Long dev cycles**

**Overwhelming operations**

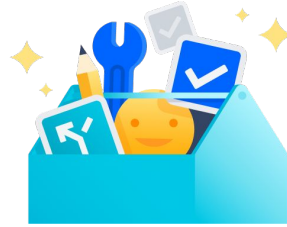
**Discoverability issues**

# Adapting to scale reliably



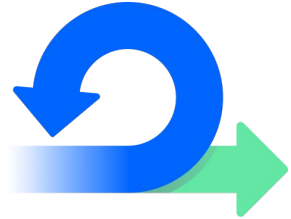
## People

- Strong team alignment



## Technology

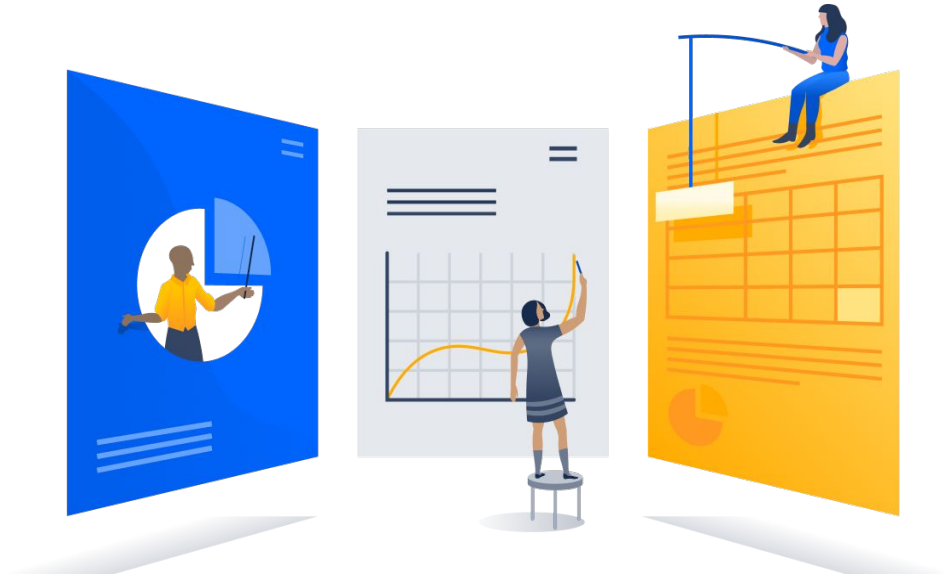
- Self-serve platforms
- Automations



## Process

- SW Dev practices
- Operational best practices

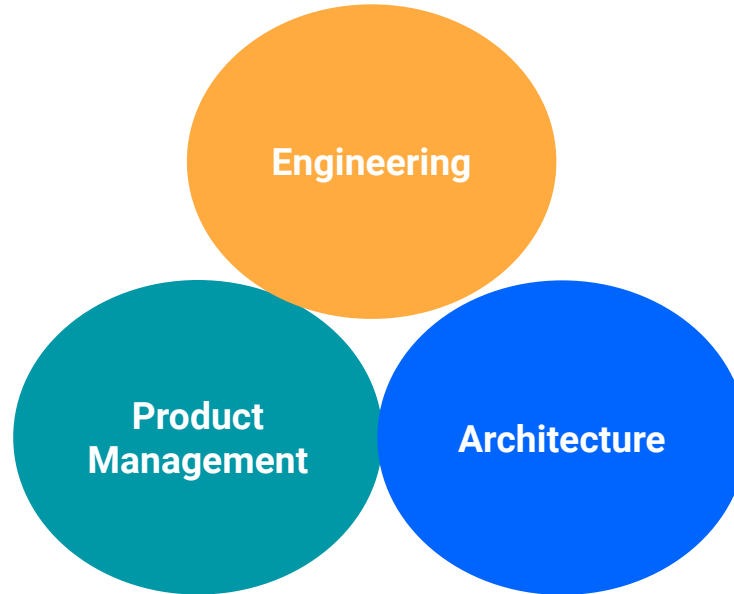
# People





# Triads

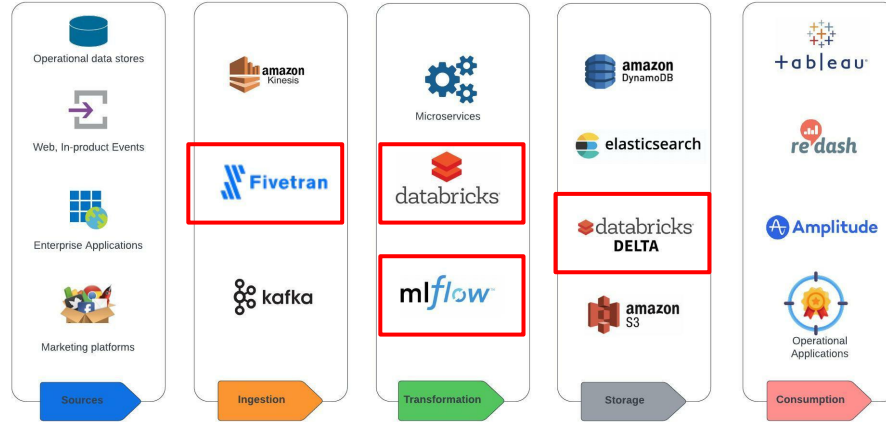
---



# Technology



# Technical Architecture



## Planning & Design

Jira Work Management Trello Confluence

## Dev & Orchestration

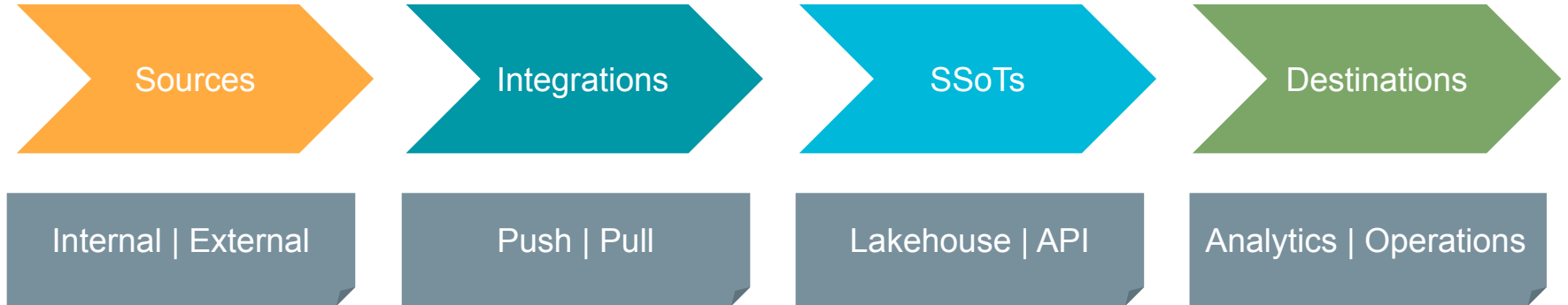
Bitbucket Apache Airflow Sourcetree

## Operations

Opsgenie Statuspage Halp **splunk**>

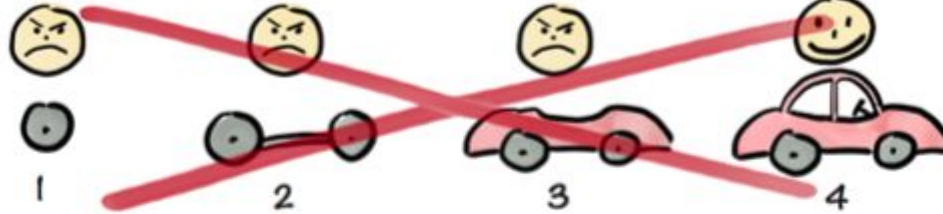
# High Level Data Flow

---

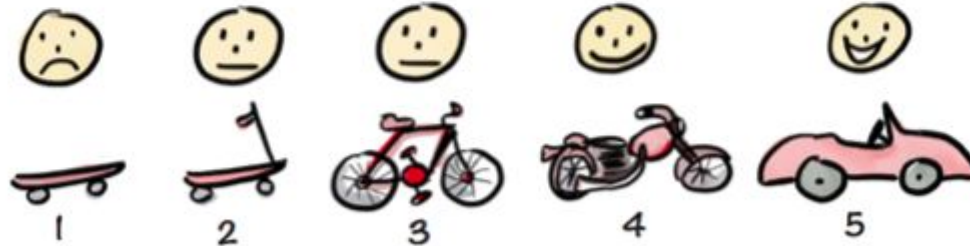


# Skateboards Vs Cars

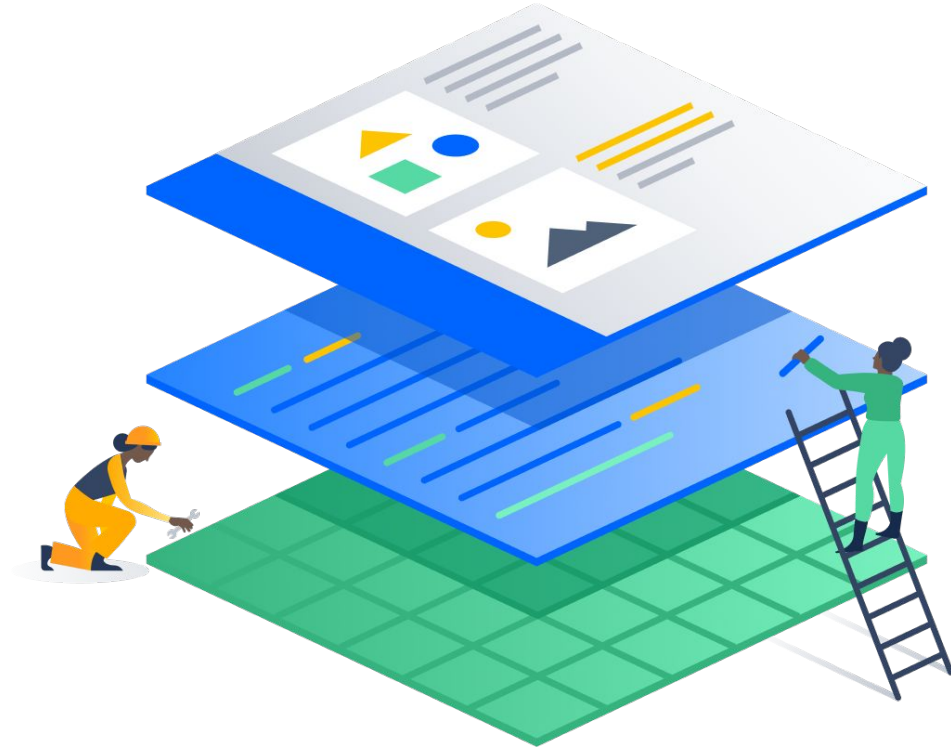
Not like this....



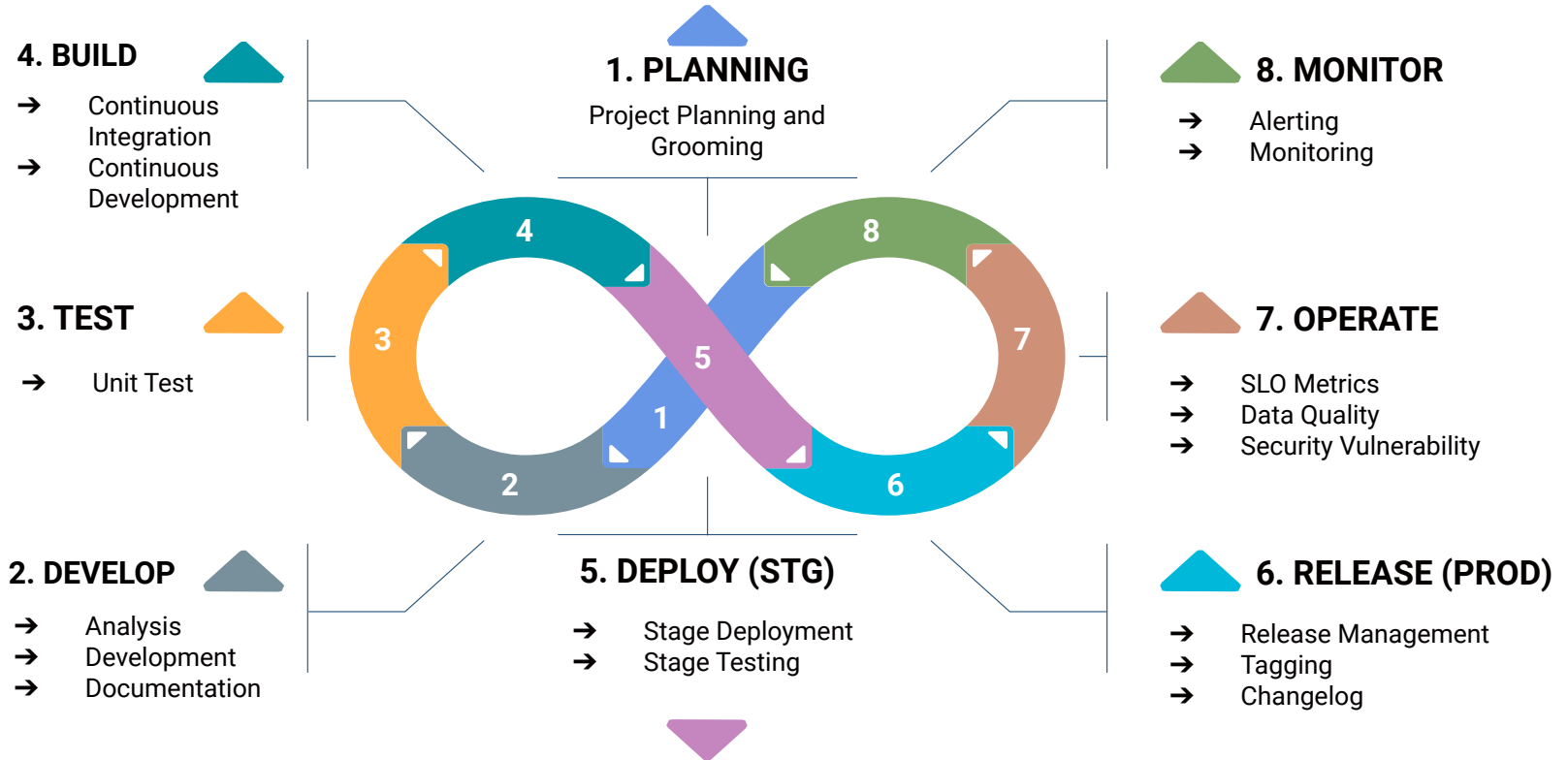
Like this!



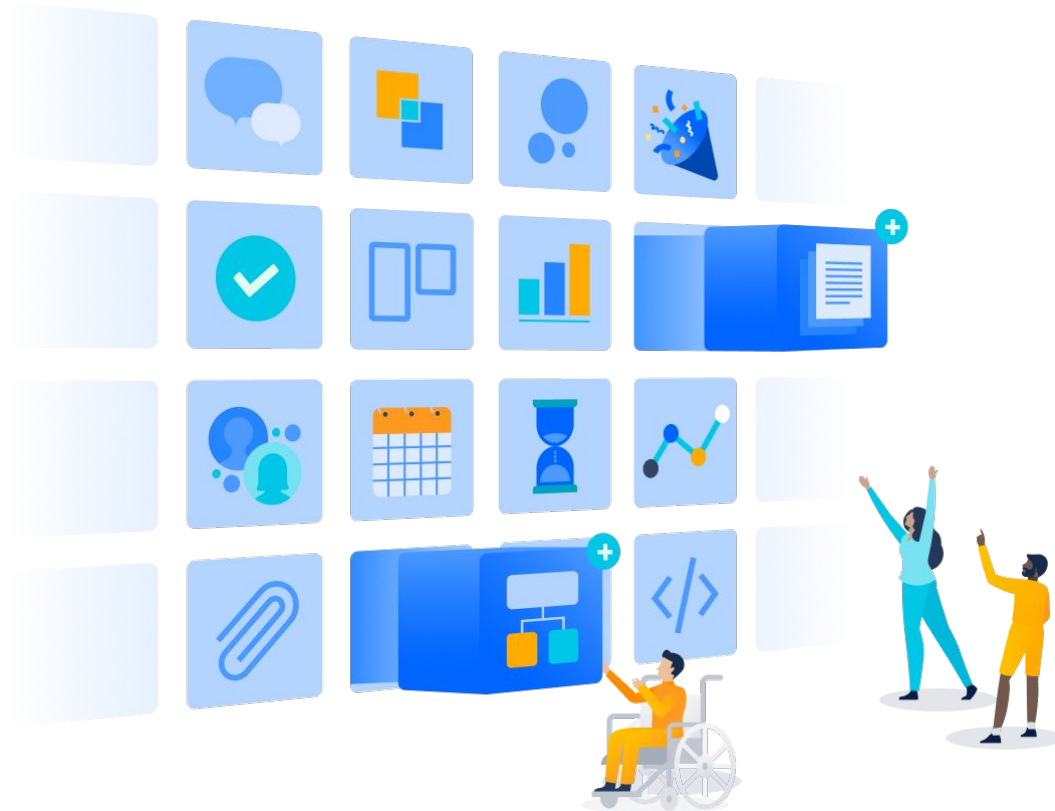
# Process



# Agile Workflow

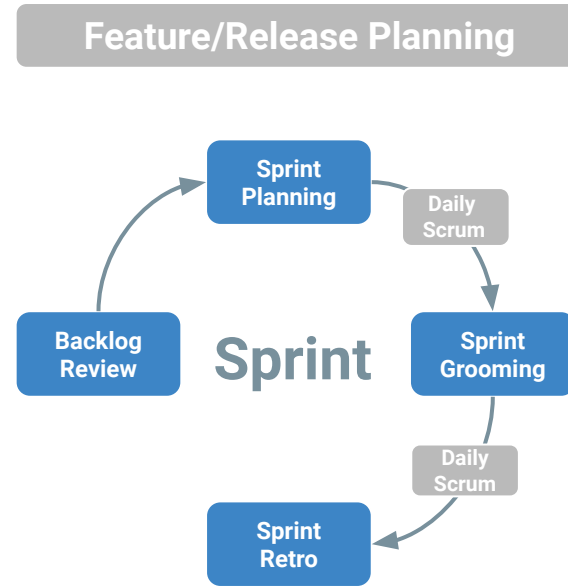
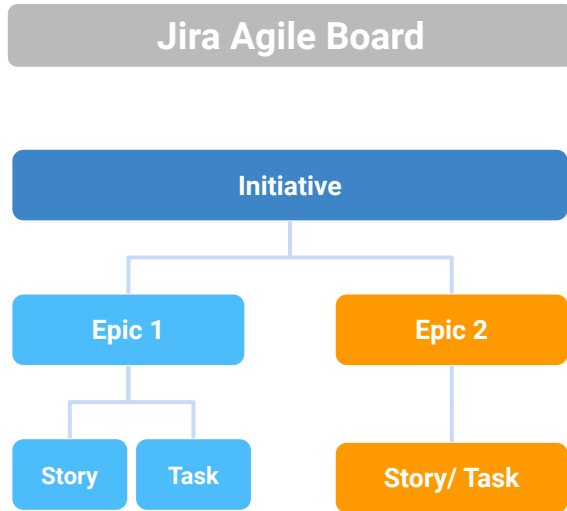


# Planning and Development

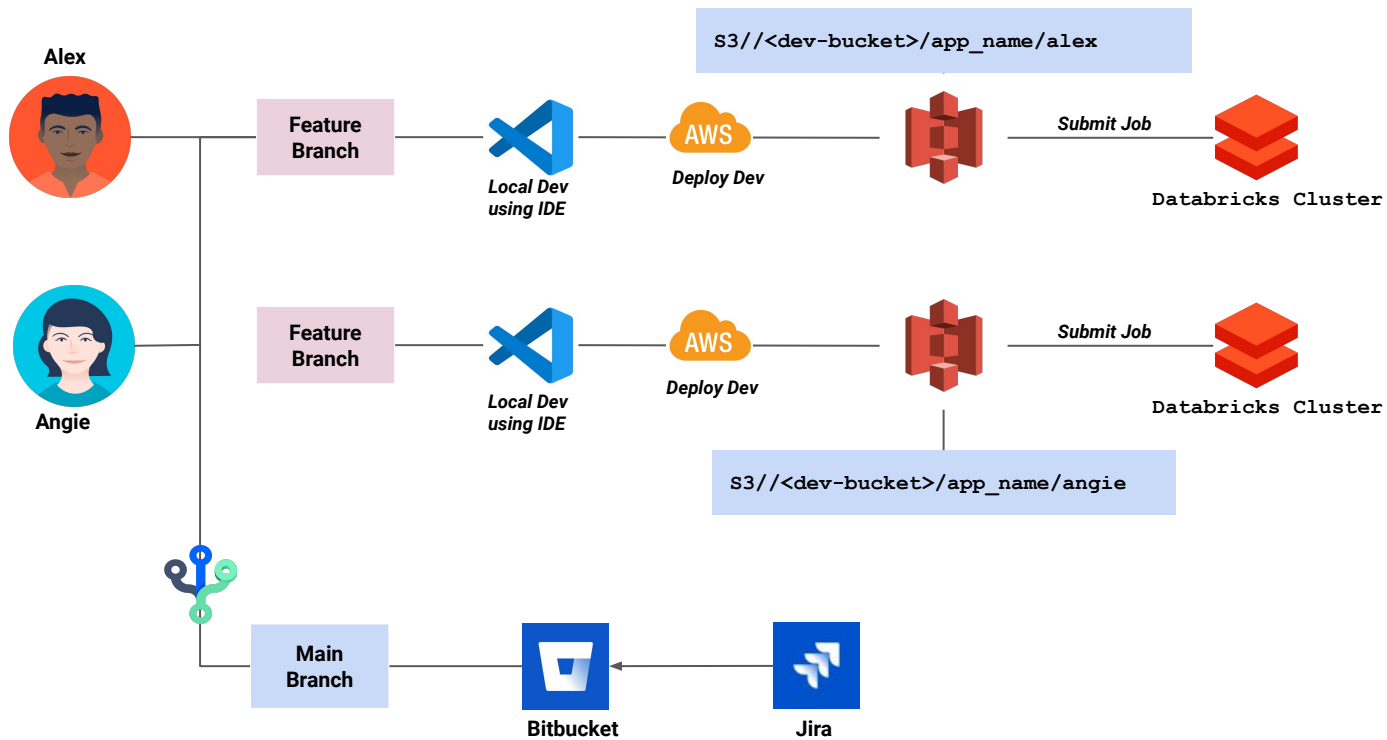




# Project Planning



# Development



# Testing

---

Dev

## Unit Test

- Compare & validate
- Generate expected & actual results
- Test and verify

Stage

## Performance Test

- Pipeline runs within expected threshold
- Resource usage is optimal
- Identify data skew joins

## Integration Test

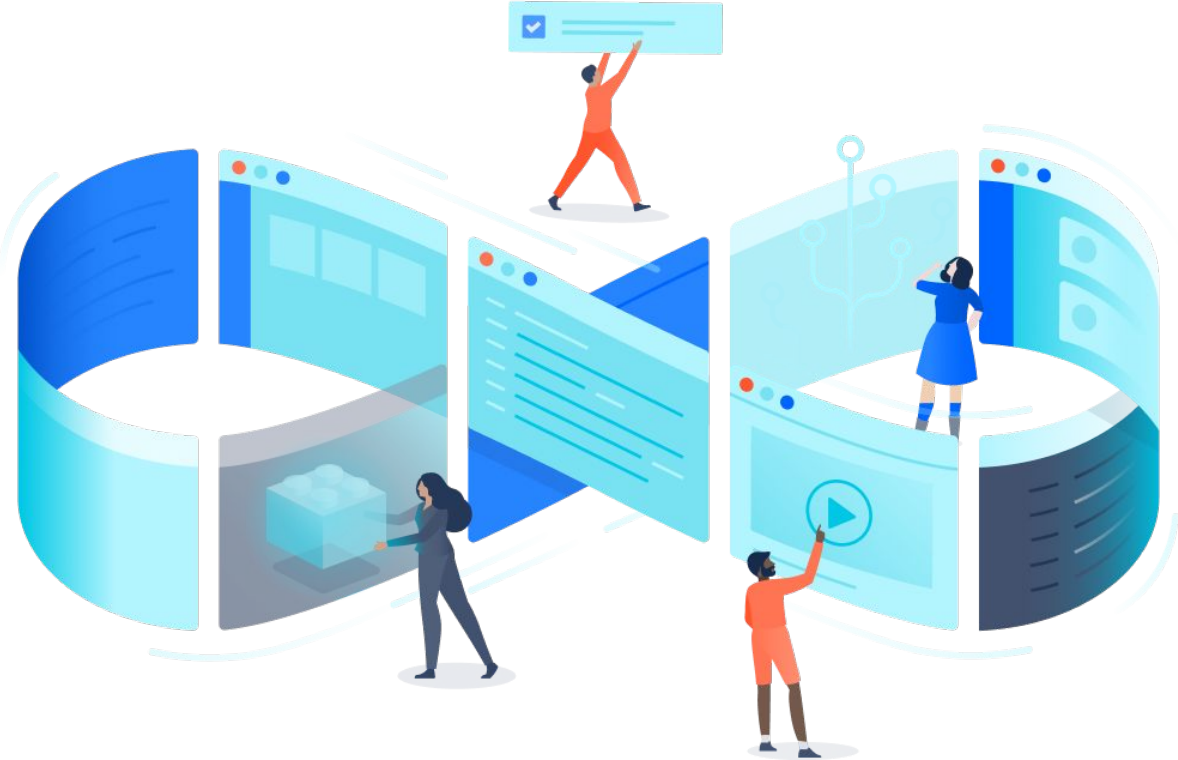
- Perform data load in lower environment
- Trigger downstream dependent tables

## User Acceptance

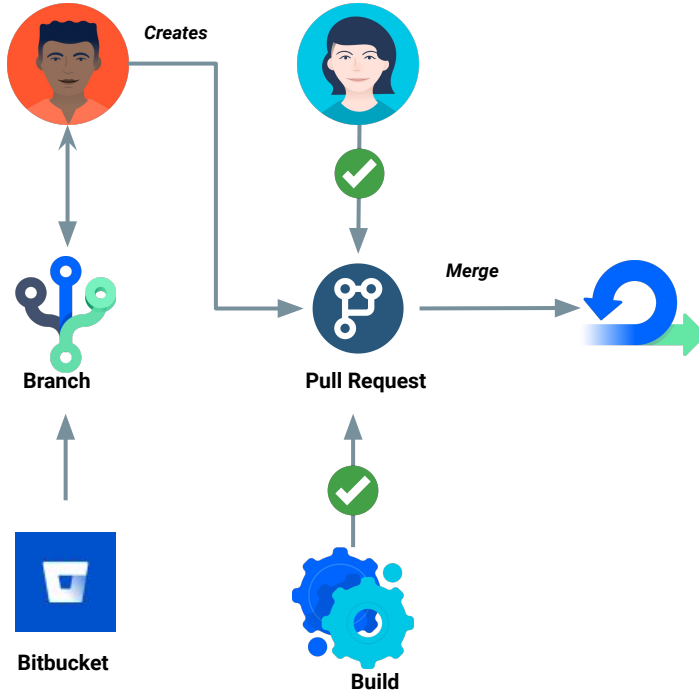
Users & downstream customers perform data validation and tests



# Continuous Integration / Development



# Build & Deploy



Pipelines:

Branches:

Main:

Parallel:

```
- step:  
  name: Test Documentation  
  scripts:  
  
- step:  
  name: Run tests  
  scripts:  
  
- step:  
  name: Deploy to staging  
  scripts:  
    - make install  
    - make build  
    - make deploy stg
```

# Release & Change Management



Creates Release

```
git checkout main
git pull
make release

  ./bin/git_next
  Deleted branch release/next
  Switched to a new branch
  'release/next'
  npx standard-version
  ✓ outputting changes to CHANGELOG.md
  ✓ committing CHANGELOG.md
  ✓ tagging release v1.6.0

git push --follow-tags origin release/next
```

## Changelog

All notable changes to this project will be documented in this file. See [standard-version](#) for commit guidelines.

### 1.6.0 (2022-06-09)

#### Features

- JIRA-1234: convert abc table to use databricks delta ([abcdef](#))

### 1.5.4 (2022-06-02)

#### Features

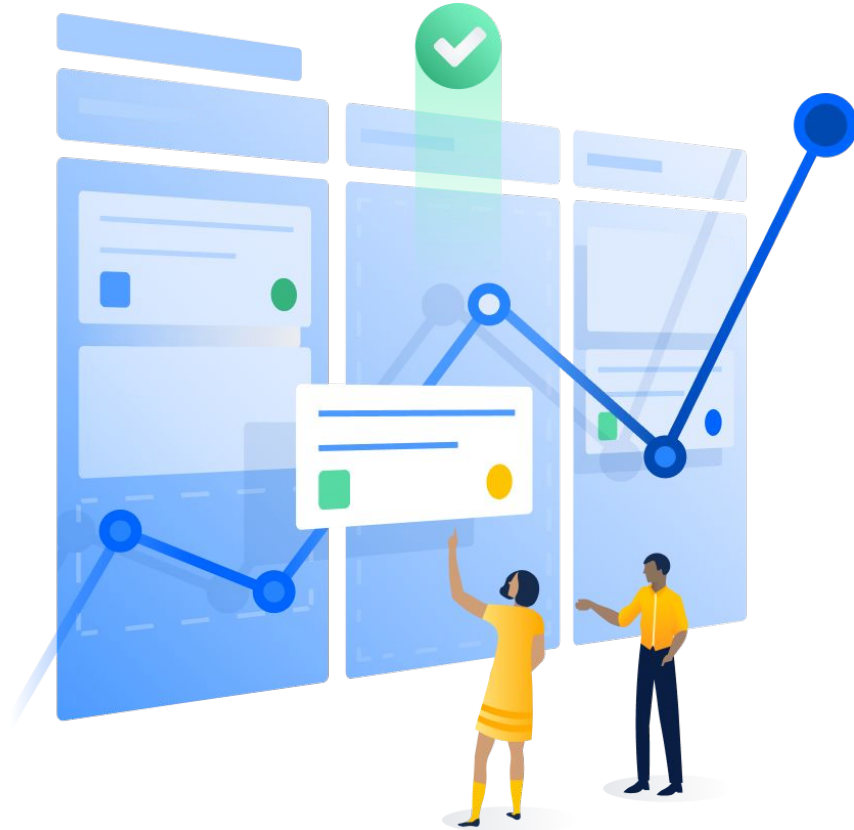
- JIRA-6789: onboard new event table ([efghij](#))

#### Bug Fixes

- JIRA-1123: Fix for multiple partition writes ([rtyght](#))
- JIRA-1991: Fix column datatype in abc table ([xtgtyy](#))
- JIRA-66179: Resolve Vulnerability in code ([kmnopq](#))

### 1.5.3 (2022-05-26)

# Data Operations



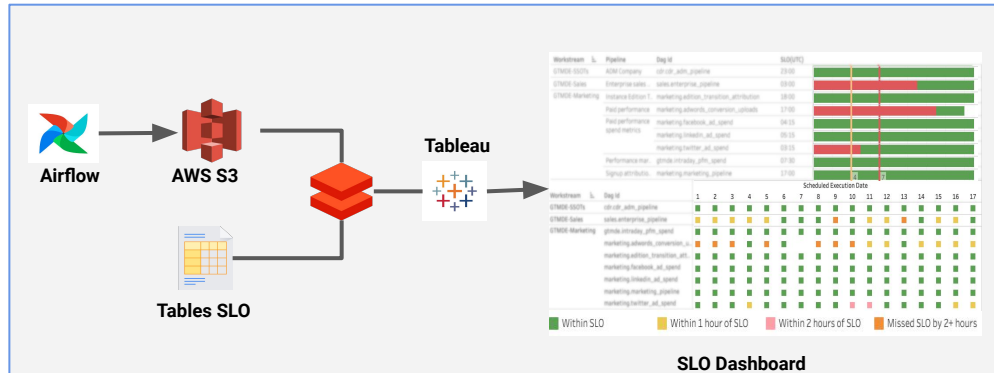
# SLA / SLO / SLI

## Data Availability & Reliability Metrics

Define

Measure

Track





# Data Quality

---



## Yoda

SQL based in-house DQ framework

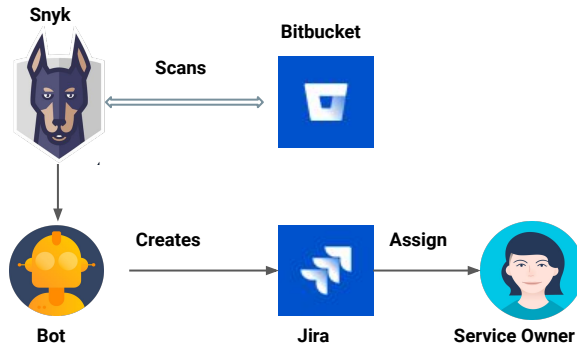


## Anomaly Detection

Prophet (open source) based time series model

# Security Vulnerabilities

---



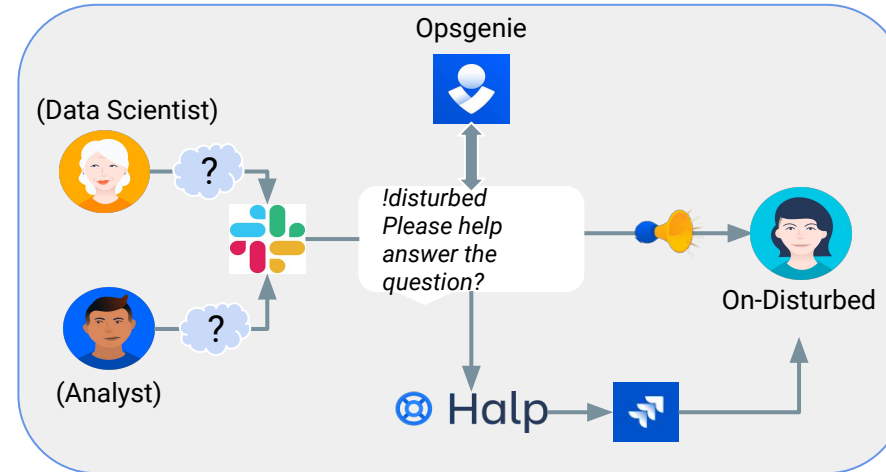
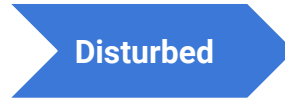
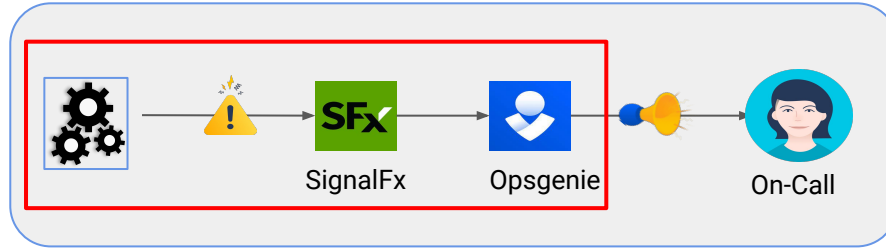
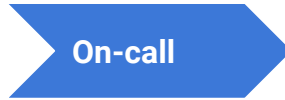
Automated Snyk Scanner

Bitbucket



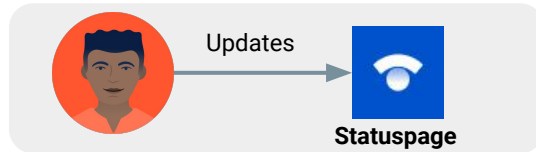
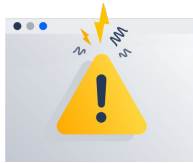
Bitbucket Recommendations

# Monitoring & Support Roles



# Incident Management

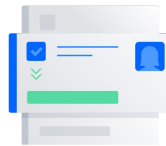
---



Statuspage updates



Incident ticket and collaboration



Post incident review process

# Team Health

---



Auto-generated Report

**SLI/SLO breaches**

Service / pipeline breaches and action items

**Vulnerability funnel**

Critical vulnerabilities with due dates

**Incidents / Post review**

Post incident review due dates

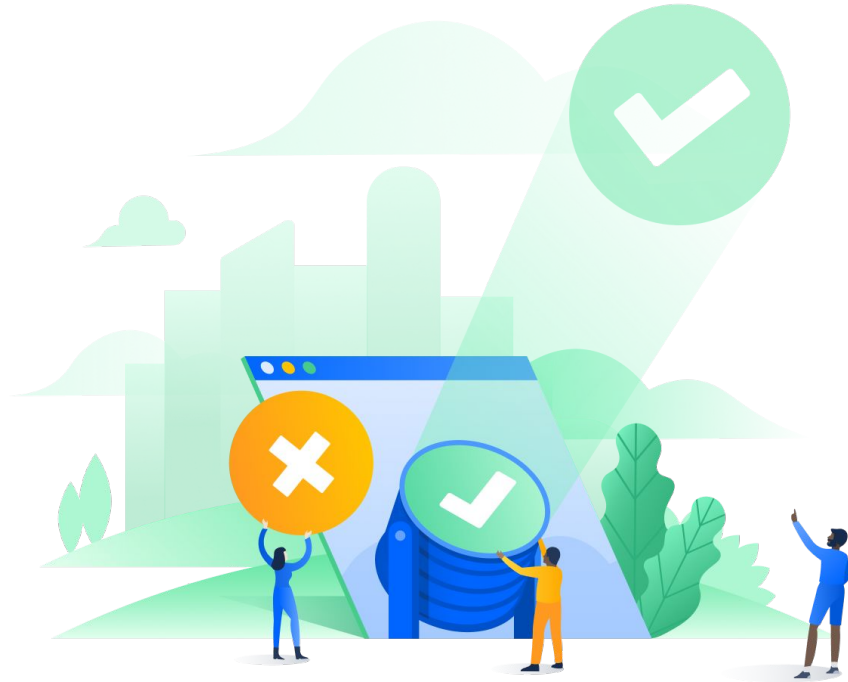
**Service/Pipeline alerts**

Alert metrics on frequency and volume

**Resource usage & cost**

Alerts for % change in cost

# Summary



# Wins

**Quicker Insights**

Dev cycles and cost reduced by 30%

**Reliability**

Incident frequency reduced by 50%

**Operations**

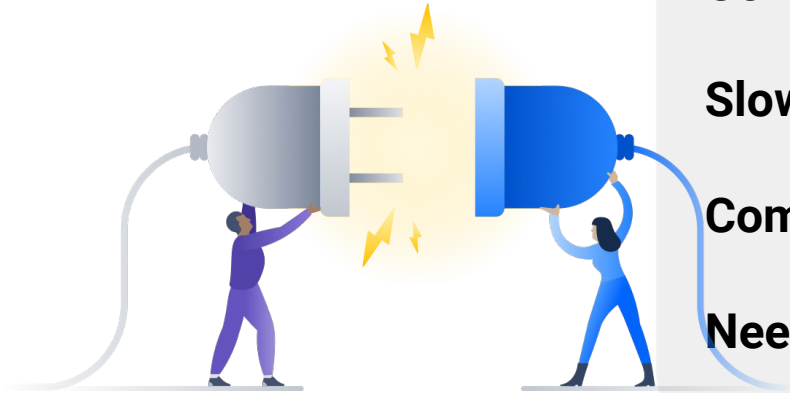
Distributed workloads

**Discoverability**

Centralized metadata, up-to-date  
documentation



# Current Challenges



**Centralization bottlenecks**

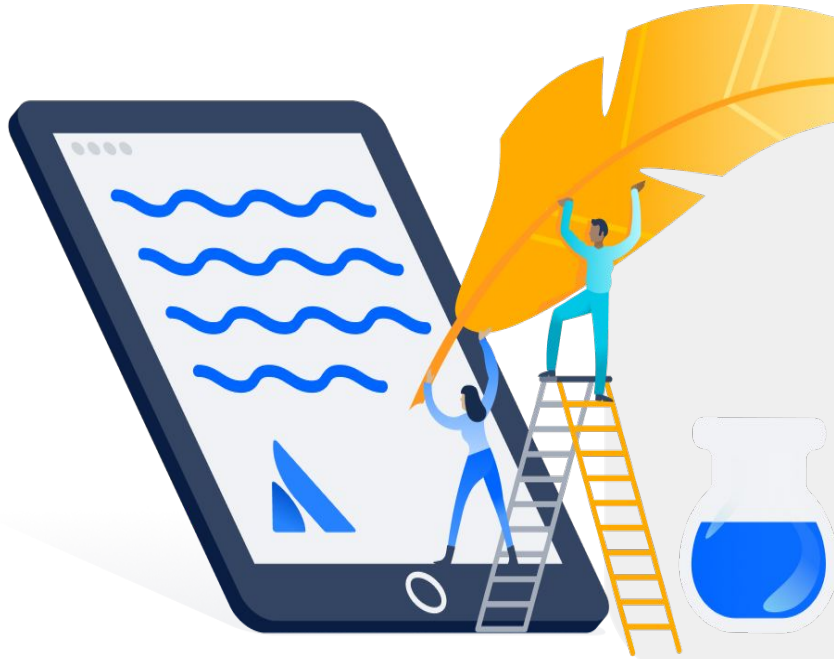
**Slower change velocity with dependencies**

**Communication and collaboration learning curve**

**Need for more training and education**



# Key Takeaways



**Scale: Databricks, Centralization, Fivetran, Workato**

**Adaptation: Team alignment and empowerment, skateboards vs cars, agile practices**

**Reliability: Tools, distributed and proactive roles**

**Continuous learning**



Thank you!



# Feedback

Your feedback is important to us  
Don't forget to rate and review the sessions