GitLab

Why won’t anyone use our pipeline??

Agenda

- Introduction
- “The Pipeline”
- Types of Jobs
  - Creative Jobs
  - Compliance Jobs
- Aligning Goals

Introduction

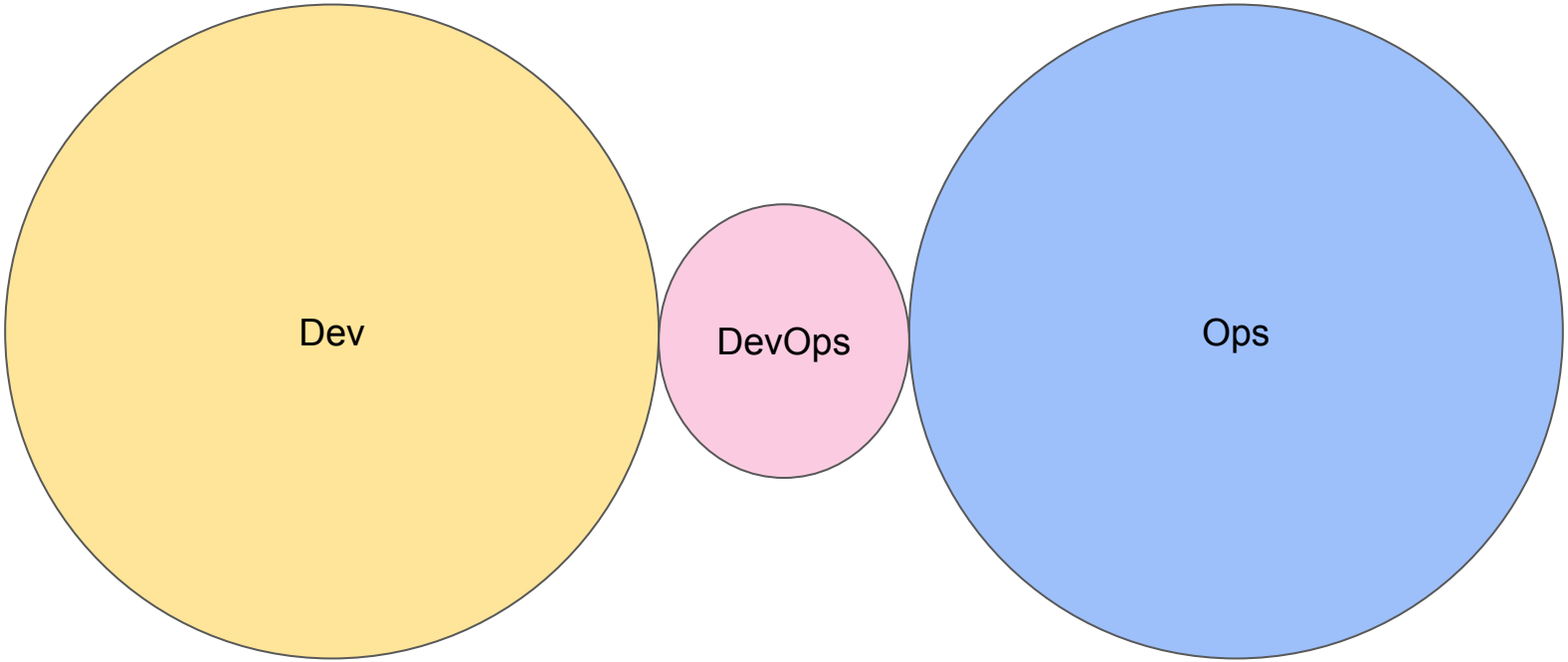
WHO

- Mike Terhar
- Technical Account Manager at GitLab
- 25 years of technology experience
  - Sysadmin
  - Developer
  - Project manager
  - Project manager manager
  - Application security architect
- For
  - Public sector (Commerce and DoD)
  - Financial institutions
  - Consultancy
  - Media

WHAT?

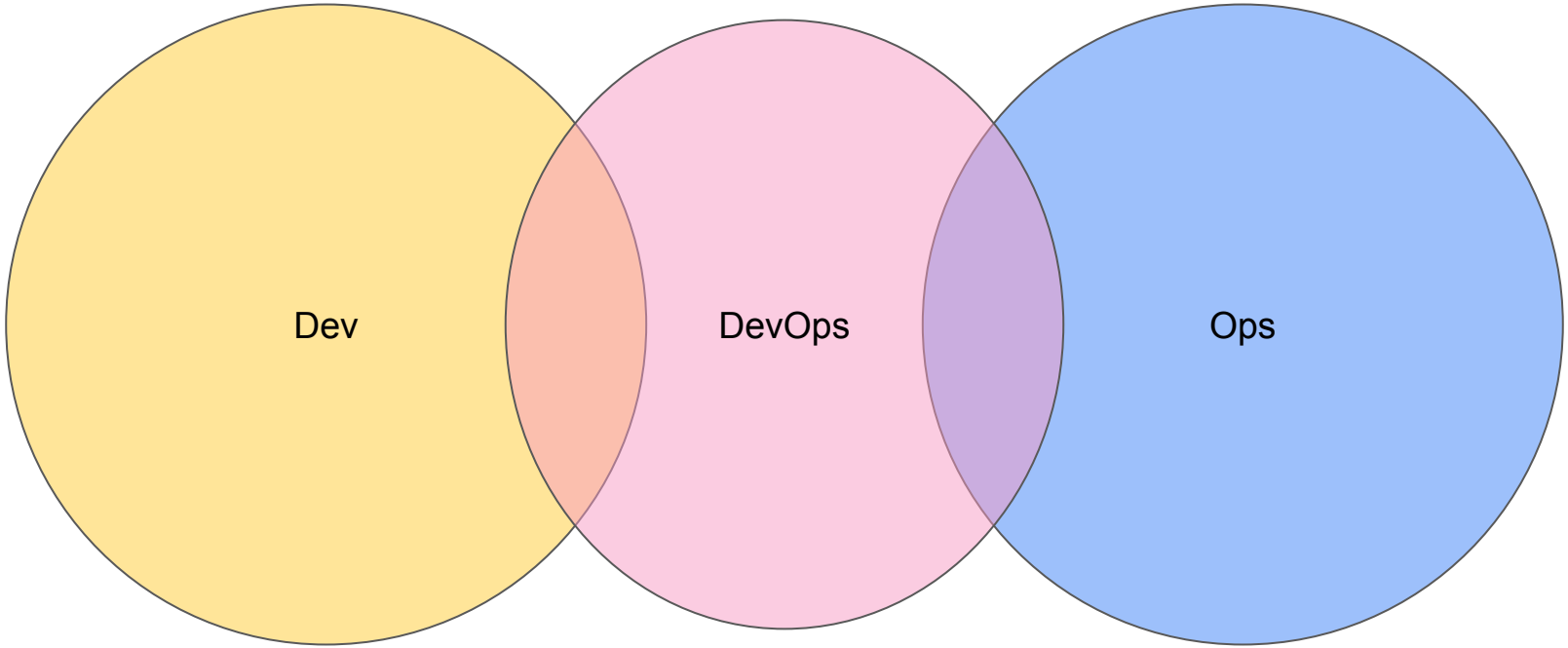
- Continuous Integration Pipelines
- Struggles
  - DevOps in compliance
  - Old and huge legacy apps
  - Security and compliance
- Best Practices
  - What are they?
  - How to foster them?
- How to do scalable DevOps
  - Onboard yourself
  - Pipeline yourself
  - ATO yourself

DevOps Pattern



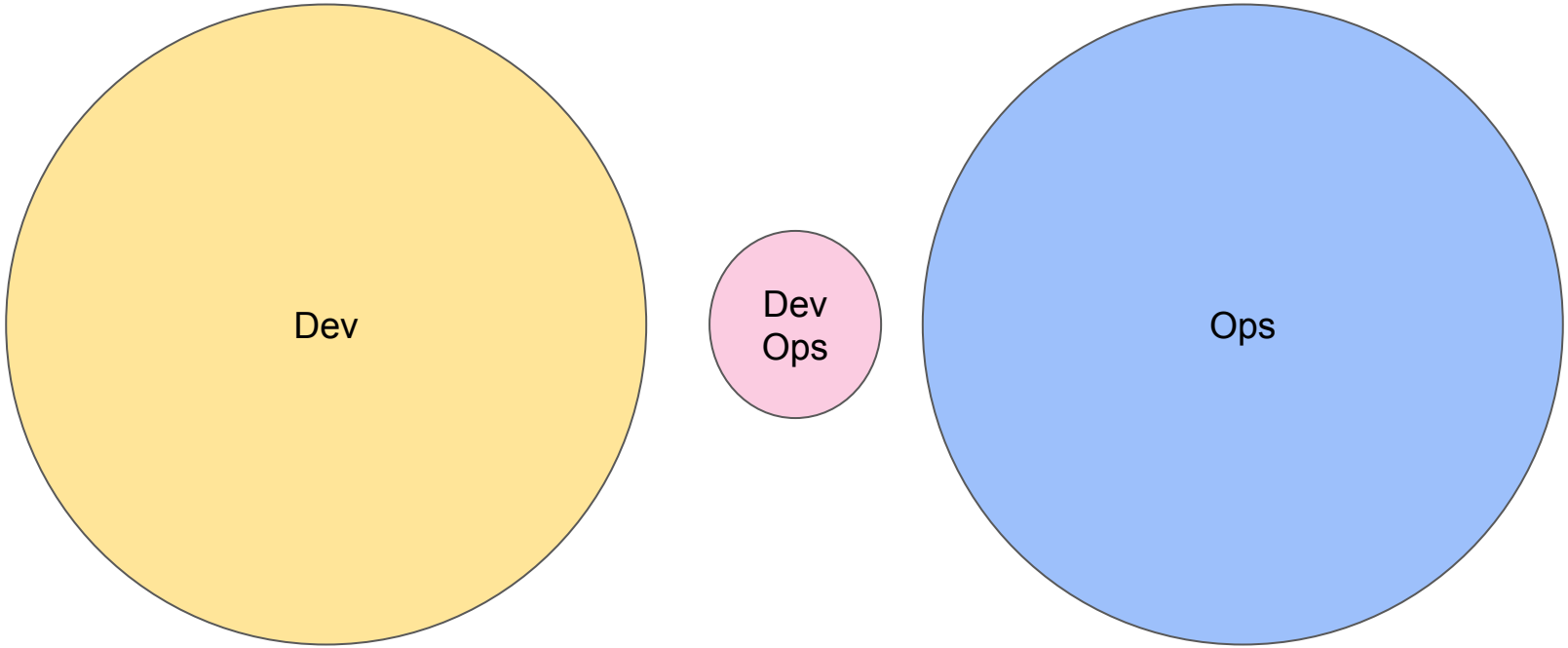
DevOps team starts between Dev and Ops, uses waterfall process and best practices to make “the pipeline”.

DevOps Pattern



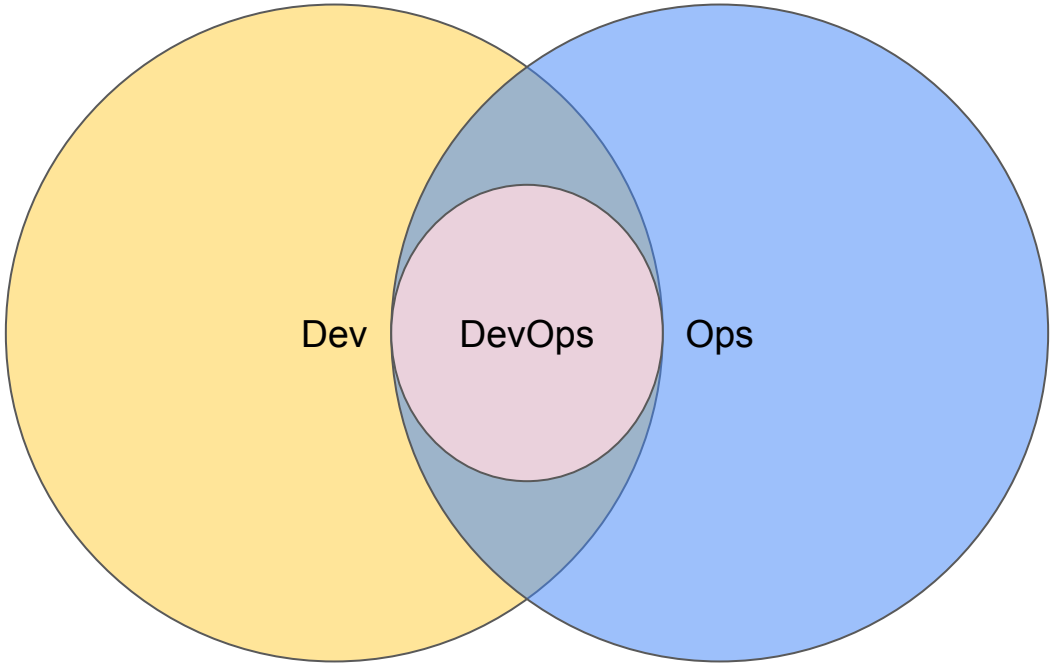
DevOps team swells because they own the pipeline and are responsible for all the jobs for all the projects.

DevOps Pattern



Or, more likely, nobody is using the pipeline because it doesn't work with the existing applications.

DevOps Pattern



DevOps team focuses on collaboration, does not own pipeline, provides job recipes, templates, and infrastructure.

Split the Pipeline

Developer / Creative Pipeline

1. Build
2. Unit Test
3. Unit Test Coverage
4. Code Quality
5. Security Scan (informational)
6. Package for deployment (container)
7. Deploy container for review

Freedom

Compliance / Enforcement Pipeline

1. Security scan (for compliance)
2. Functional tests (IV&V)
3. Deploy to Production

LOCKED DOWN

Split the Projects

Developer / Creative Project

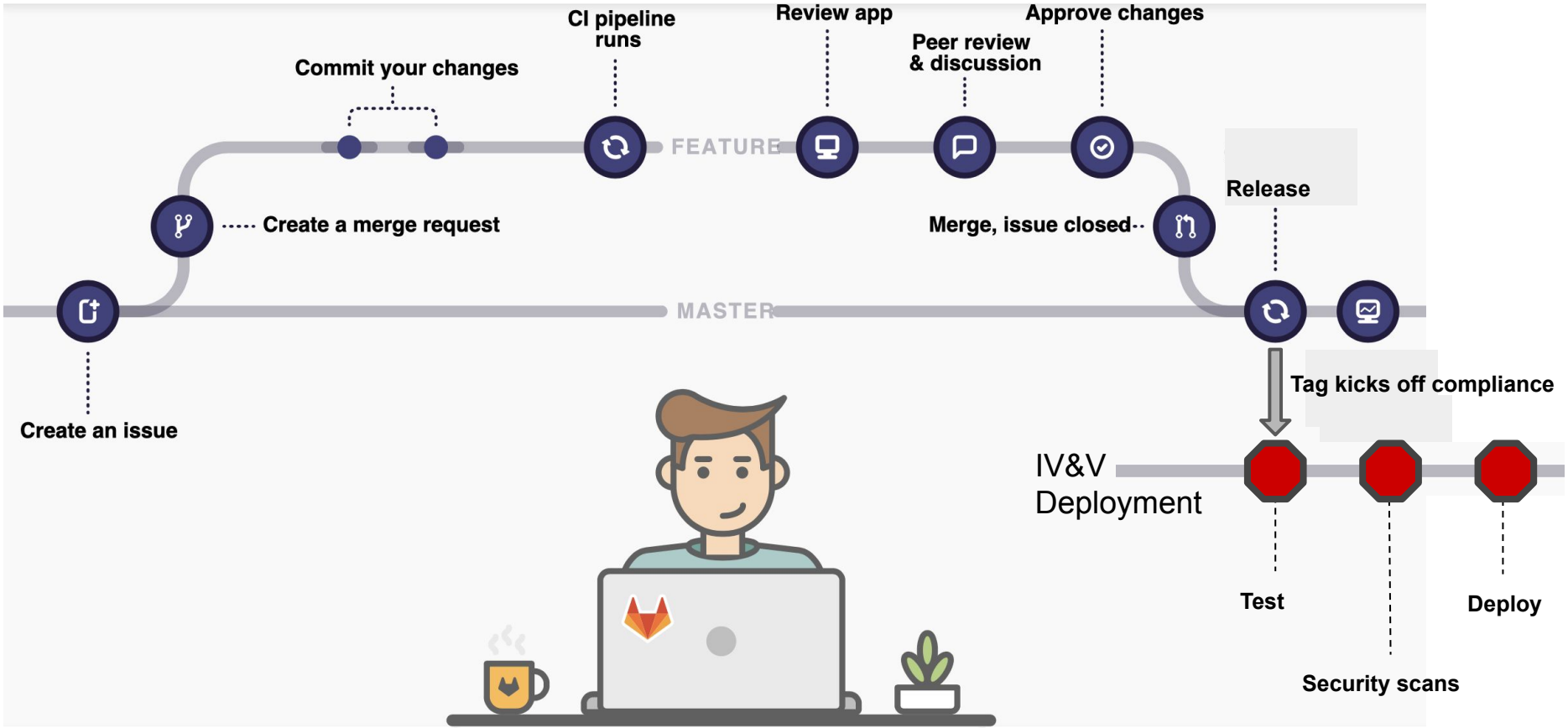
1. Issues
2. Code
3. Pipeline
4. Artifacts
5. Containers
6. Review/Dev environments

Compliance / Enforcement Project

1. IAC for deployment
2. Automated functional tests
3. Security scan (for compliance)
4. Manual test cases
5. Deploy to Production

Developer pipeline triggers IV&V pipeline when RC tag

Developer Workflow and Release Process



Main image from https://about.gitlab.com/solutions/gitlab-flow

Aligned Goals

1. Developers are empowered for velocity and quality
2. Compliance goals are met while limiting drag
3. All work is visible in GitLab

Bonus: IV&V and Ops teams are rewarded for automating tests and moving to IAC

So many more facets

This brief discussion was an introduction to a very deep and complex set of topics.

More information available at:

- Articles and examples at https://brownfield.dev
- Trunk-based workflow from https://about.gitlab.com/solutions/gitlab-flow
- Venn diagrams paraphrased from https://web.devopstopologies.com/#type-one
- “Rich Change Control with GitLab” https://www.youtube.com/watch?v=uW95PV8d-w8&t=704