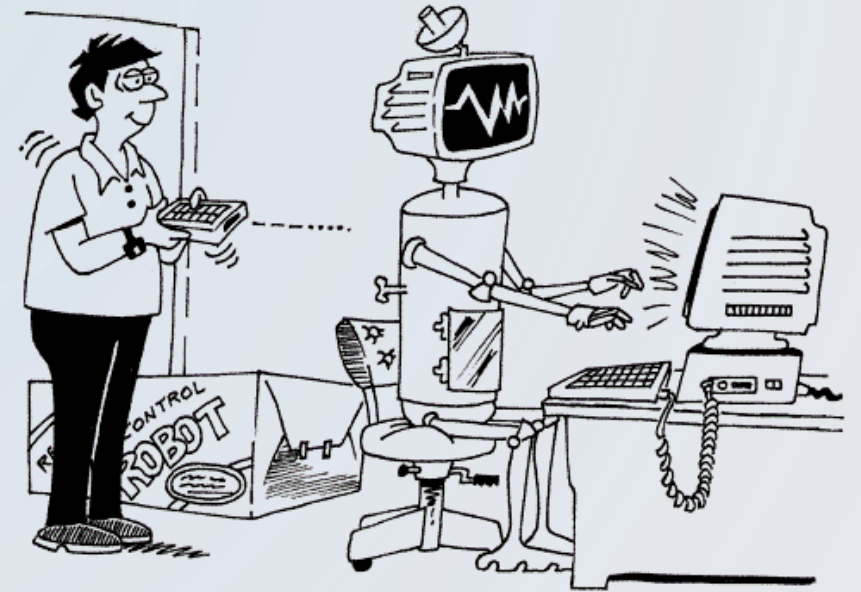# DEVOPS IN SAP

A walk through of an attempt to structure a Continuous Integration(CI) pipeline around Fiori & ABAP development.

A brief discussion of the tools and how they can be deployed as ad-hoc microservices when the pipelines need them.

DEVELOPMENT LIFECYCLE

The textbook Software Development Lifecycle

But what about...

- User stories or requirements documentation?

- Functional and technical documentation?

- Development standard compliance?

- Code Inspection or performance analysis?

- Code readability / pretty print?

- Build / Compile - In the SAP world this would be... generate ABAP, minify Fiori app, etc

- Unit, integration, and regression testing?

TOOLS

Q: What tools are out there to help automate?

Code Management	Job Runner	Code Checking	ABAP Testing	Test Recording	Web Testing
Gitlab		ABAP Code Inspector	ABAPUnit		Qunit
					OPA
SubVersion	Jenkins		eCATT	Katalon	Selenium
Github	Travis CI	Jslint / Grunt	Postman (ODATA API Testing)	Selenium IDE	
Transports		WebIDE			UIVeri5 ? Protractor ? Puppeteer ?

PIPELINES

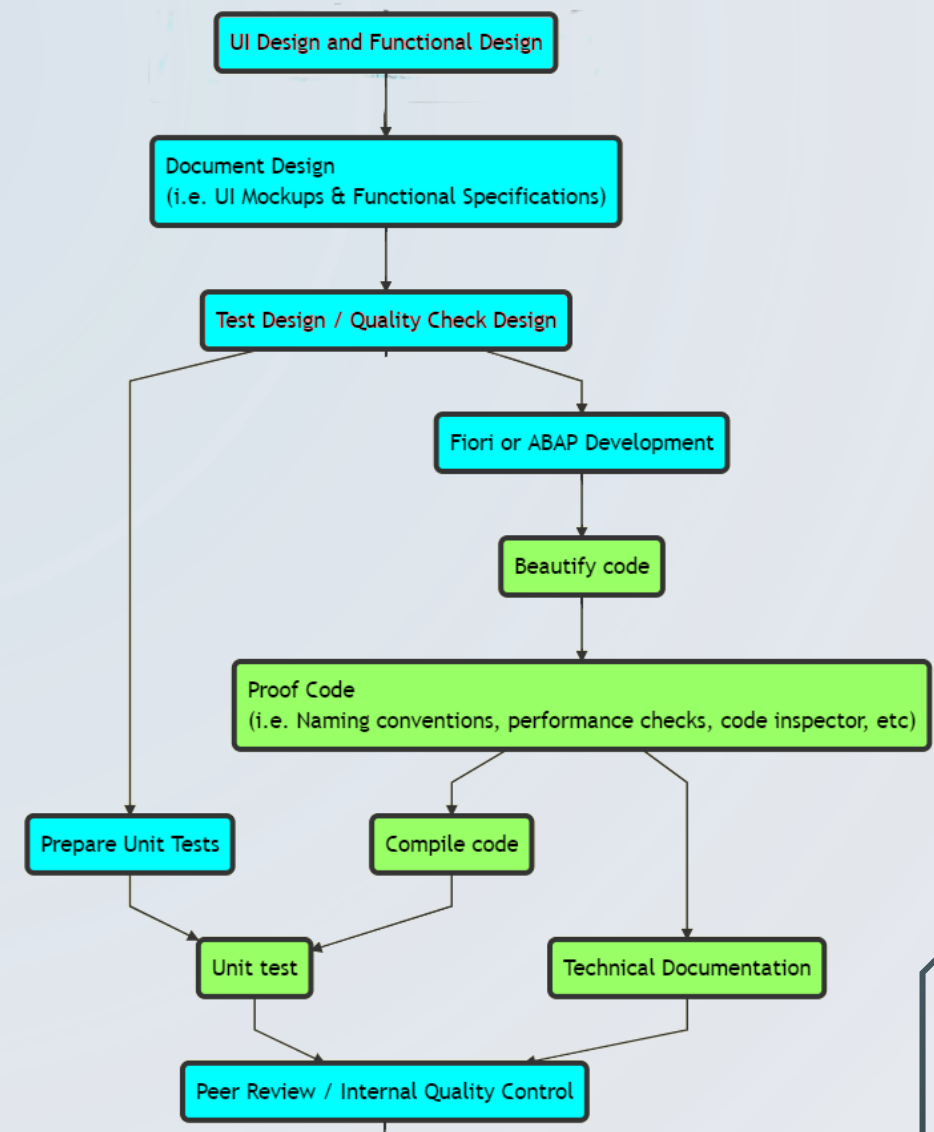Q: What do we mean by a pipeline?

A: Arranging your flow of work so that you can hand-off to a chain of automated activities and let the system do work for you.

Example:

Key

- Critical thinking activity (i.e. Human-centric)
- Can be automated

DEVOPS

Q: What is DevOps?

A:

"A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality"
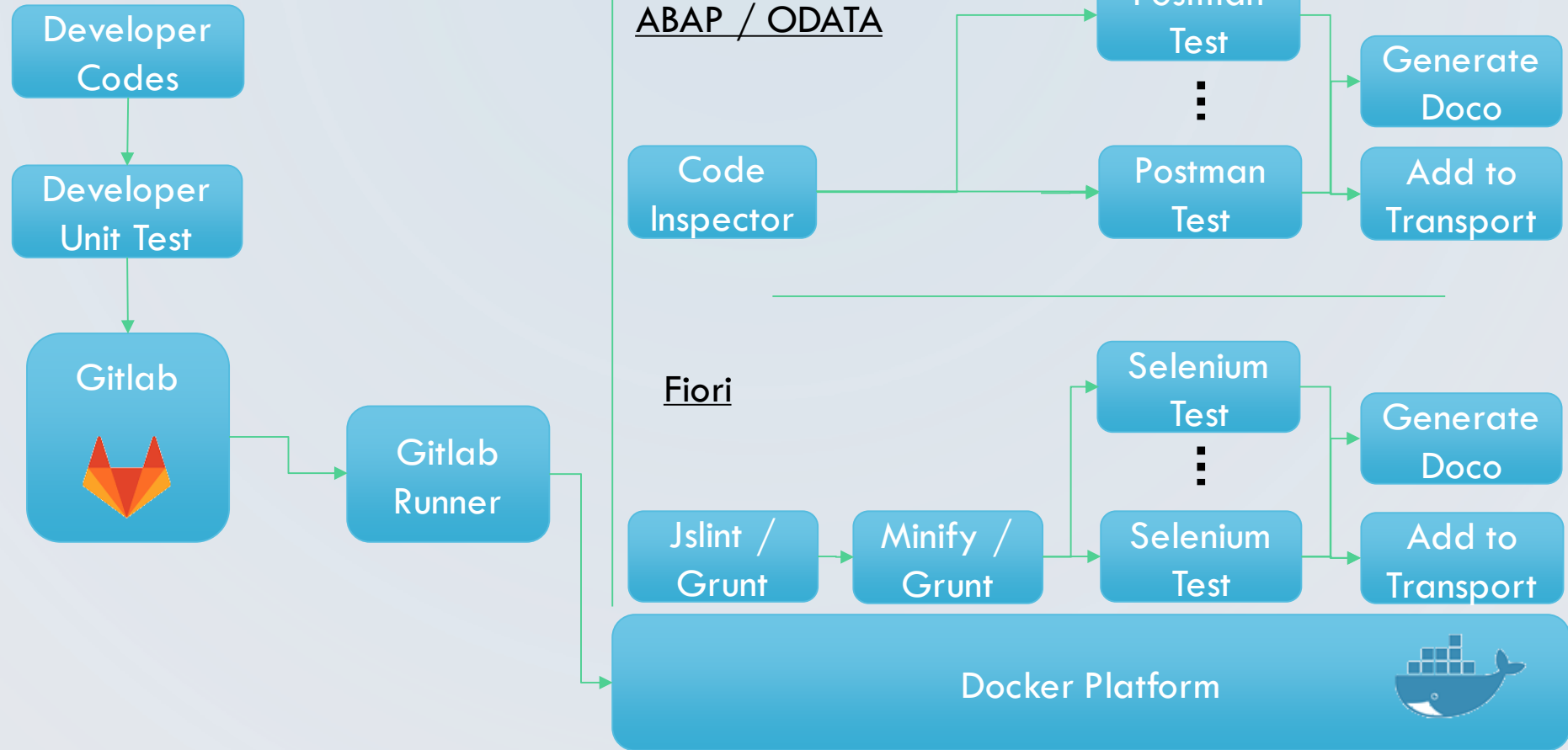
<div align="right">Software Engineering Institute</div>

The lay-person explanation:

DevOps is a portmanteau of the words **Dev**elopment and IT **Op**erations.

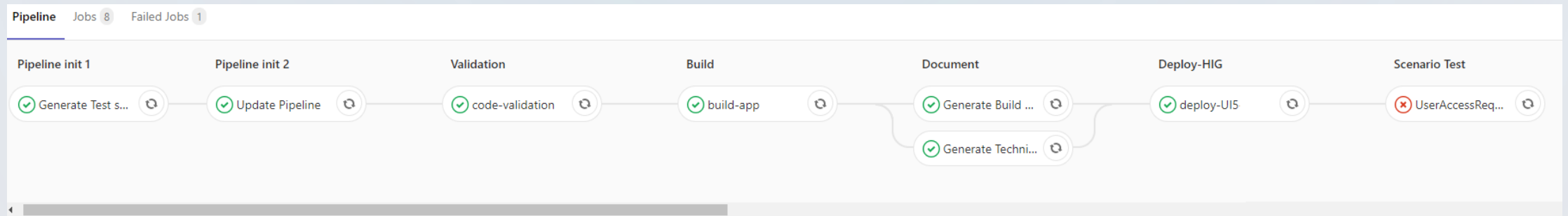Basically, coding/automating of how you build and deploy software.

The Quality part comes from reliability improvements gained from being able to repeat things to high quality every time due to automated checks & balances.
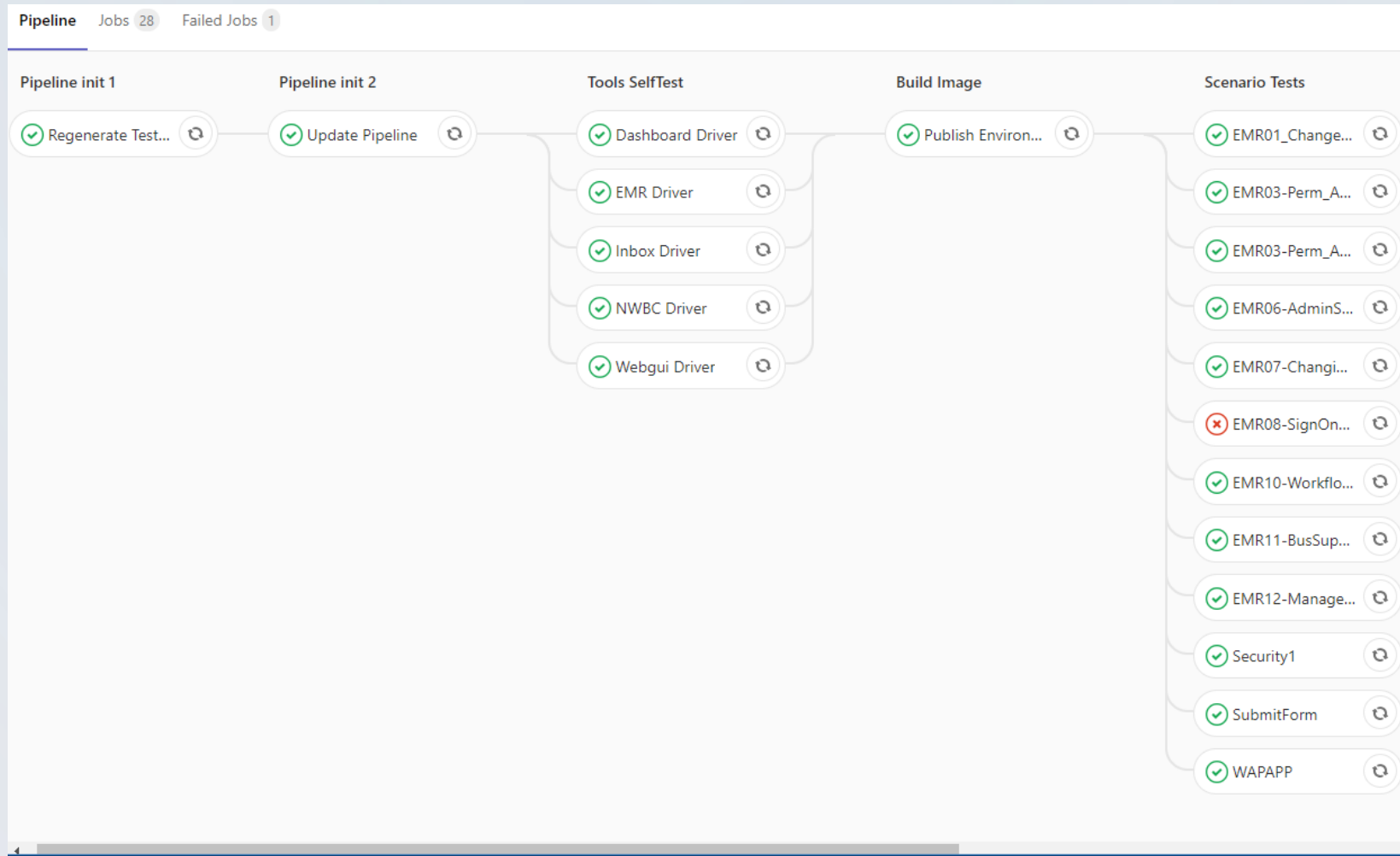
OUR SETUP

PIPELINES IN GITLAB

Pipeline on each App for CI:

PIPELINES IN GITLAB

Nightly Regression Test Everything:

HOW DID IT WORK OUT?

The Challenges:

- **Learning Curve** – Lots of new tools for team to learn.
- **Inclusiveness** - Bringing the whole team along for the ride, not just the developers can be a challenge.
- **IT barriers** - it can be difficult to get hardware for a "non-basis" team when the specification for the hardware is just "Cloud". It can be difficult to install extra software on desktops in large corporate environments.

The Benefits:

- **Repeatability** – lowered cost and increased speed of updates or patching (Critical business processes can be regression tested quickly, changes can be tried and rolled back quickly).
- **Confidence** – being able to run some broad tests every time meant we could say with confidence we hadn't broken something elsewhere.
- **Quality**– less defects are raised from testing, and less fixes sent back for further changes. Delivered a higher quality product.

WANT MORE SESSIONS LIKE THIS?

Matt Coote	matt+insidetrack@grayleigh.net
Jakob Kjaer	jakob+insidetrack@agilux.com.au