Continuously Adding value via CI/CD

by
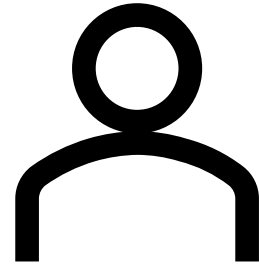
Ahmed M. Eisa

- **Our pains /reasons**
- **Solutions**
- **Continuous integration**
- **Continuous deployment**
- **What is takes ?**

Agenda

Pains /reasons

Human

Consumes time

Makes errors

Revenue

- Returning to a working state on failure

- Time to deliver new feature

- Down time, for switching versions, failures to be noticed

Costs

Security holes
Security testing takes time

Infrastructure errors
Mistakes on deployment

Unused cloud resources
Left some time while switching production

Merging features issues
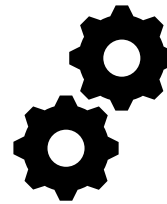Extra time added to resolve them

Bugs in production
Manual testing takes time

Solution

Revenue

- Returning to a working state on failure (protecting revenue)

- Time to deliver new feature (more features quickly) more revenue

- Down time, for switching versions, failures to be noticed (protecting revenue)

Automation

Reduces time
Preventing errors

Security holes
Security testing run automatically

Infrastructure errors
No human errors,

Unused cloud resources
Automated cleanup run immediately, reduces cost
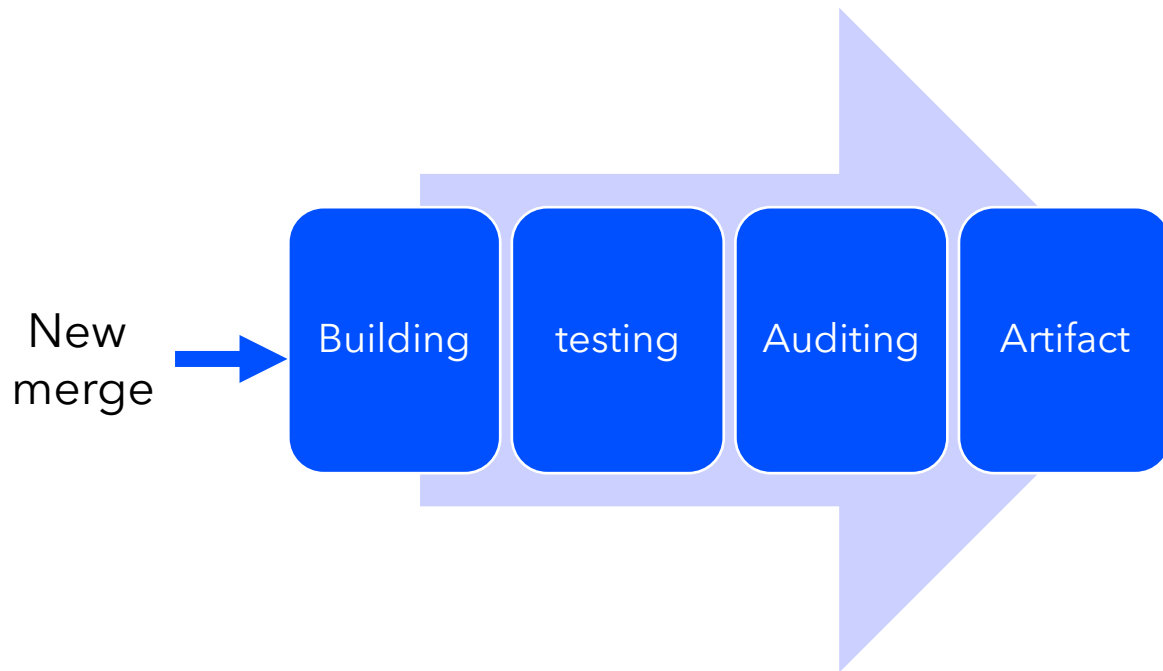
Merging features issues
Quick fixes due to automated quick alerts

Bugs in production
- Automated tests in lower time
- Fixing bugs before production costs much less than after production

Costs

Continuous integration

New
merge → **Building** **testing** **Auditing** **Artifact**
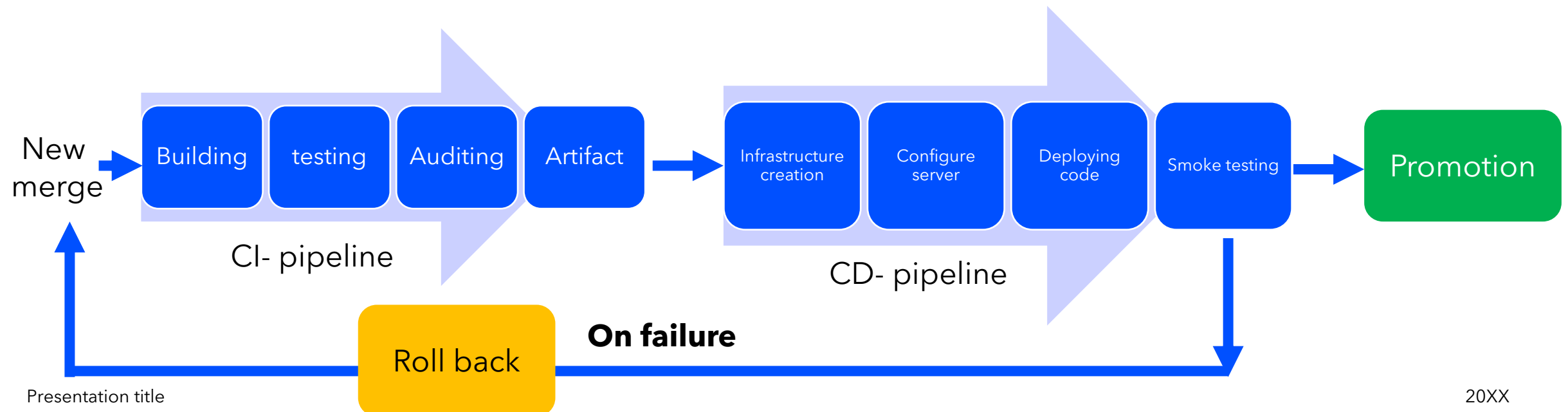
(Artifact making part)

Merging different branches of different features done by different developers is a tedious task and unmanageable unless we have a shared pipeline to combine all merges of these branches

In this pipeline features are frequently and continuously merged in one shared place and processed automatically

Continuous deployment

(Introducing the artifact)

Repeatable and tedious tasks of creating infrastructure , configuring them and deploying the artifacts could be automated in one delivery pipeline that ends with a smoke test to rollback quickly and save costs. In this way we can continuously add value to customer without introducing deployment errors to them

New merge → **Building** — **testing** — **Auditing** — **Artifact** →
CI- pipeline

Infrastructure creation — Configure server — Deploying code — Smoke testing → **Promotion**
CD- pipeline

Roll back — **On failure**

What is takes ?

- Automate everything
  - No manual deployment
  - No manual testing
  - No manual server configurations
  - No manual auditing
- Learning and applying new skills to automate tasks.