

The Fundamentals of CI/CD

CONTINUOUS INTEGRATION



The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact! Some common CI-related phases might include:

- Compile
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact



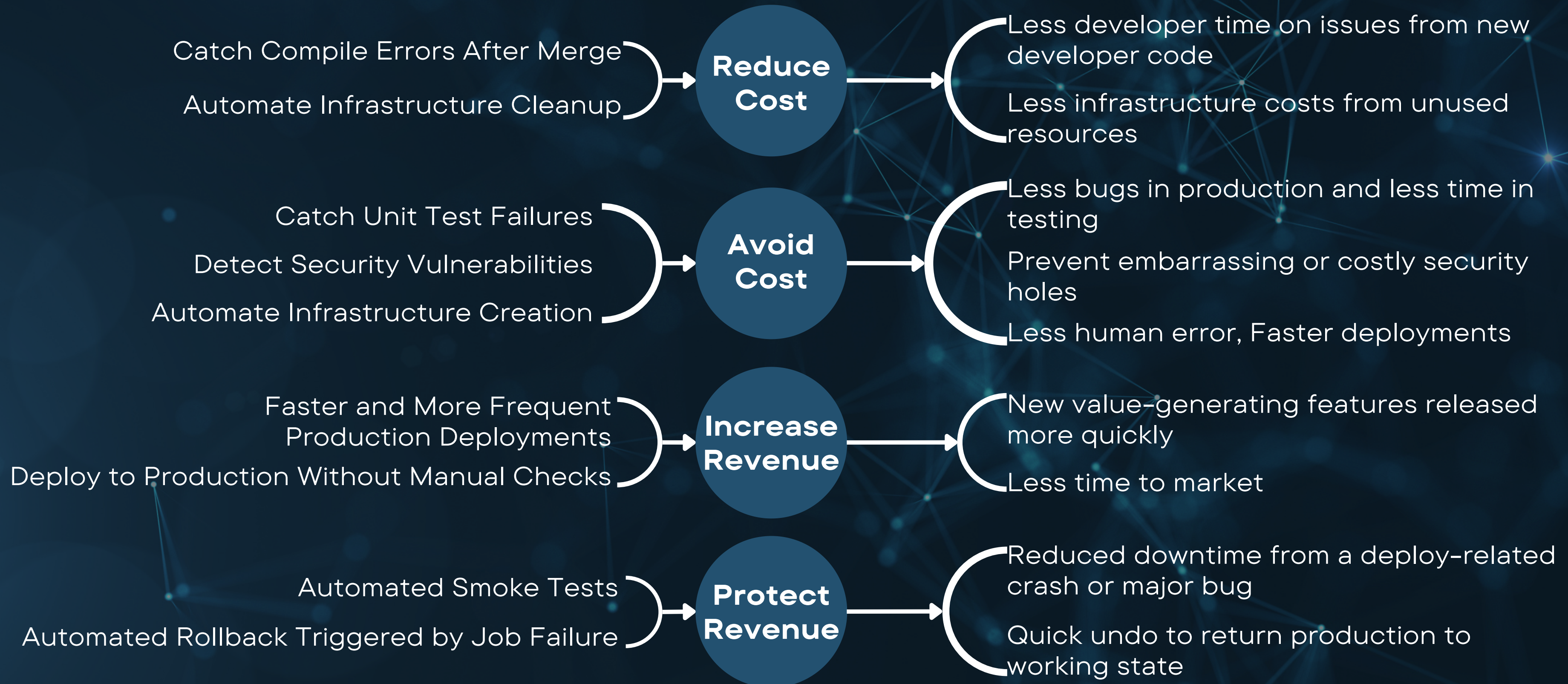
CONTINUOUS DEPLOYMENT

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "Moving" the artifact from the shelf to the spotlight. Some common CD-related phases might include:

- Creating infrastructure
- Provisioning servers
- Copying files
- Promoting to production
- Smoke Testing (aka Verify)
- Rollbacks



Benefits of CI/CD



Where Does CI/CD Fit In?

Before CI/CD

Human

Human, Subjective, Inconsistent

Human

Human

Human, Hit or Miss, Easily Bought Off with Pressure

Human, Hit or Miss, Easily Bought Off with Pressure

Human,
Often Not Done

Human, Problematic,
Missed Steps

Coding

Code Review

Compile/Lint

Merge/Integrate

Run Unit Tests

Run Integration Tests

Verify Dependency Security

Deploy to Test Env

After CI/CD

Human

Human/CI - Static Analysis

CI

CI

CI

CI

CI

CD

Cont...

Before CI/CD

Human, Time Consuming

Human, Problematic,
Missed Steps

Human, Often Unnecessary If Pre-
Development Activities are on Point

Human, Problematic,
Missed Steps, Stressful

Human, Problematic,
Missed Steps, Stressful

Human, Inconsistent

Human, Problematic,
Missed Steps, Stressful

Human, Problematic,
Missed Steps, Stressful

After CI/CD

Team Test

CD - Automated Acceptance Tests

Deploy to Client Test Env

CD

Client Test

Human, Maybe Not Needed If
We Can Build Confidence

Create Infrastructure

CD

Deploy to Production

CD

Smoke Test in Prod

CD

Rollbacks

CD

Promoting Production

CD

The Setback of CI/CD



- No more manual deploying to environments
- No more modifying environment settings in GUI's
- No more neglecting the unit tests
- No more leaving broken code in place
- Requires a high level of discipline
- Requires additional skills to maintain and extend automation

