The Fundamentals of CI/CD

CONTINUOUS INTEGRATION

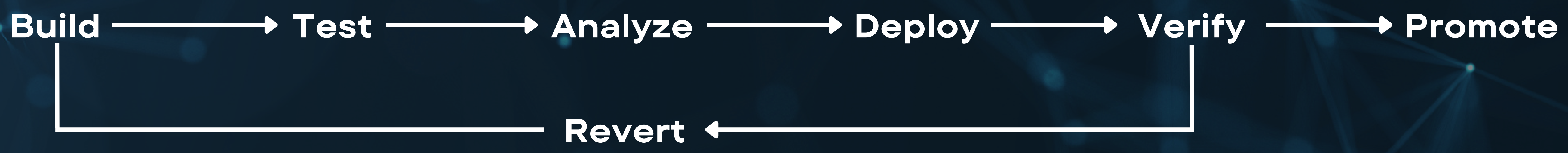
The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact! Some common CI-related phases might include:

- 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



Build → Test → Analyze → Deploy → Verify → Promote

Revert

Where Does CI/CD Fit In?

Before CI/CD		After CI/CD
Human	Coding	Human
Human, Subjective, Inconsistent	Code Review	Human/CI - Static Analysis
Human	Compile/Lint	CI
Human	Merge/Integrate	CI
Human, Hit or Miss, Easily Bought Off with Pressure	Run Unit Tests	CI
Human, Hit or Miss, Easily Bought Off with Pressure	Run Integration Tests	CI
Human, Often Not Done	Verify Dependency Security	CI
Human, Problematic, Missed Steps	Deploy to Test Env	CD

Cont...

Before CI/CD		After CI/CD
Human,Time Consuming	Team Test	CD - Automated Acceptance Tests
Human, Problematic, Missed Steps	Deploy to Client Test Env	CD
Human, Often Unnecessary If Pre-Development Activities are on Point	Client Test	Human, Maybe Not Needed If We Can Build Confidence
Human, Problematic, Missed Steps, Stressful	Create Infrastructure	CD
Human, Problematic, Missed Steps, Stressful	Deploy to Production	CD
Human, Inconsistent	Smoke Test in Prod	CD
Human, Problematic, Missed Steps, Stressful	Rollbacks	CD
Human, Problematic, Missed Steps, Stressful	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

Removing Human Error

A world without Human Error is something of a myth that is something impossible to achieve since almost ***everything*** requires human intention. As of today, there have been many inventions that have been created to reduce such human errors and CI/CD is included amongsts them.