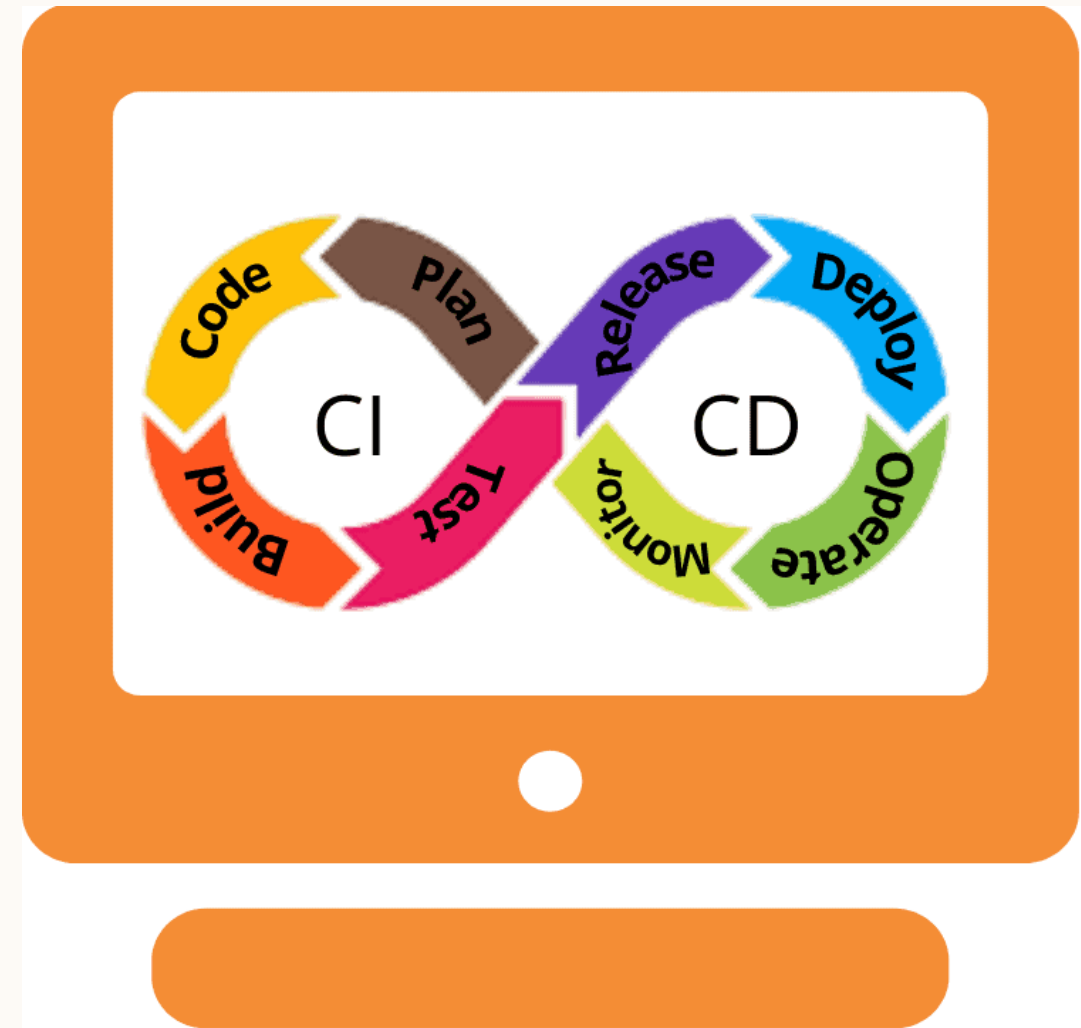


CI/CD INNOVATION

AND UDAPEOPLE PRODUCT



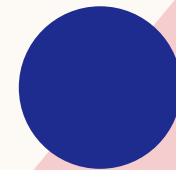
AGENDA

Introduction CI/CD

Where Does CI/CD Fit In?

Benefits of CI/CD

Best Practices for CI/CD



INTRODUCTION

In software engineering, CI/CD or CICD is the combined practices of continuous integration (CI) and (more often) continuous delivery or (less often) continuous deployment (CD)

WHERE DOES CI/CD FIT IN?

Where Does CI/CD Fit In?

Stage	Before CI/CD	After CI/CD
Coding	Human	Human
Code Review	Human, Subjective, Inconsistent	Human/CI - Static Analysis
Compile/Lint	Human	CI
Merge/Integrate	Human	CI
Run Unit Tests	Human, Hit or Miss, Easily Bought Off with Pressure	CI
Run Integration Tests	Human, Hit or Miss, Easily Bought Off with Pressure	CI
Verify Dependency Security	Human, Often Not Done	CI
Deploy to Test Env	Human, Problematic, Missed Steps	CD
Team Test	Human, Time Consuming	CD - Automated Acceptance Tests
Deploy to Client Test Env	Human, Problematic, Missed Steps	CD
Client Test	Human, Often Unnecessary If Pre-Development Activities are On Point	Human - Maybe Not Needed If We Can Build Confidence
Create Infrastructure	Human, Problematic, Missed Steps, Stressful	CD
Deploy to Production	Human, Problematic, Missed Steps, Stressful	CD
Smoke Test in Prod	Human, Inconsistent	Automated Smoke Tests (Subset of AAT's)
Rollbacks	Human, Problematic, Missed Steps, Stressful	CD
Promoting Production	Human, Problematic, Missed Steps, Stressful	CD
Celebrate!	Human	Human

BENEFITS OF CI/CD

Technical Language	Value	Translation
Catch Compile Errors After Merge	Reduce Cost	Less developer time on issues from new developer code
Catch Unit Test Failures	Avoid Cost	Less bugs in production and less time in testing
Detect Security Vulnerabilities	Avoid Cost	Prevent embarrassing or costly security holes
Automate Infrastructure Creation	Avoid Cost	Less human error, Faster deployments
Automate Infrastructure Cleanup	Reduce Cost	Less infrastructure costs from unused resources
Faster and More Frequent Production Deployments	Increase Revenue	New value-generating features released more quickly
Deploy to Production Without Manual Checks	Increase Revenue	Less time to market
Automated Smoke Tests	Protect Revenue	Reduced downtime from a deploy-related crash or major bug
Automated Rollback Triggered by Job Failure	Protect Revenue	Quick undo to return production to working state

BEST PRACTICES FOR CI/CD



FAIL FAST

Finding and investigating failures as fast as possible. The faster you can bring your code failures to light, the faster you can fix them.



MEASURE QUALITY

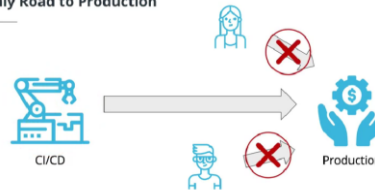
Measure your code quality so that you can see the positive effects of your improvement work

Low
Unit Test
Coverage



Low
Quality
Code

Only Road to Production



ONLY ROAD TO PRODUCTION

Any other person or process that meddles with production after CI/CD is running will inevitably cause CI/CD to become inconsistent and fail



MAXIMUM AUTOMATION

find and reveal failures as fast as possible. The faster you can bring your code failures to light, the faster you can fix them.



FAST DEPLOYMENT

So fast

SUMMARY

What are we waiting for?

CI/CD allows organizations to ship software quickly and efficiently and save the money! UdaPeople product needs it!

THANK YOU