CI/CD FUNDAMENTALS AND BENEFITS

WHY USE IT?

Ahmed Gamal Ali

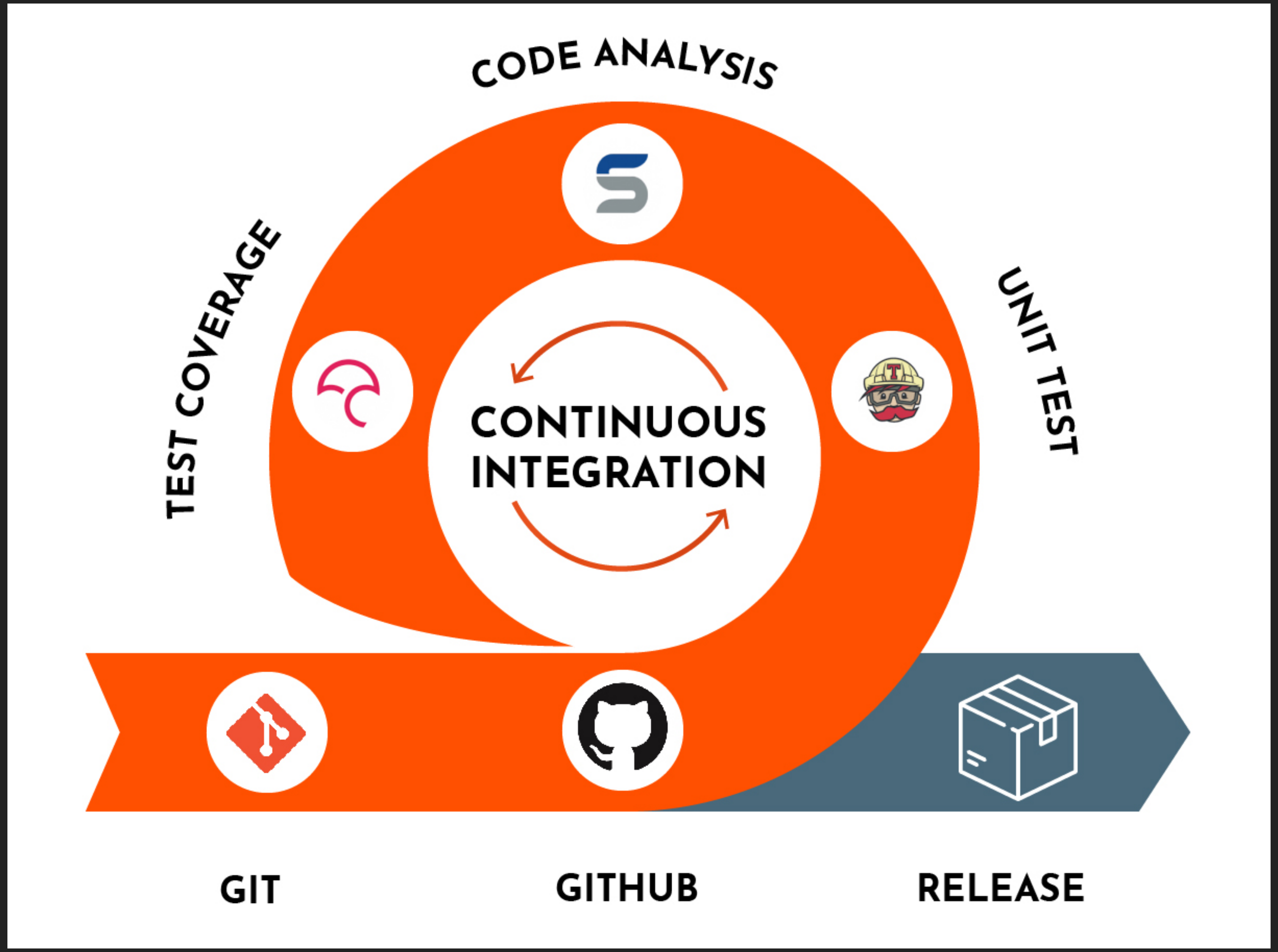
WHAT IS CI/CD?

CI/CD is the combined practices of:

- Continuous Integration
- Continuous Delivery
- Continuous Development

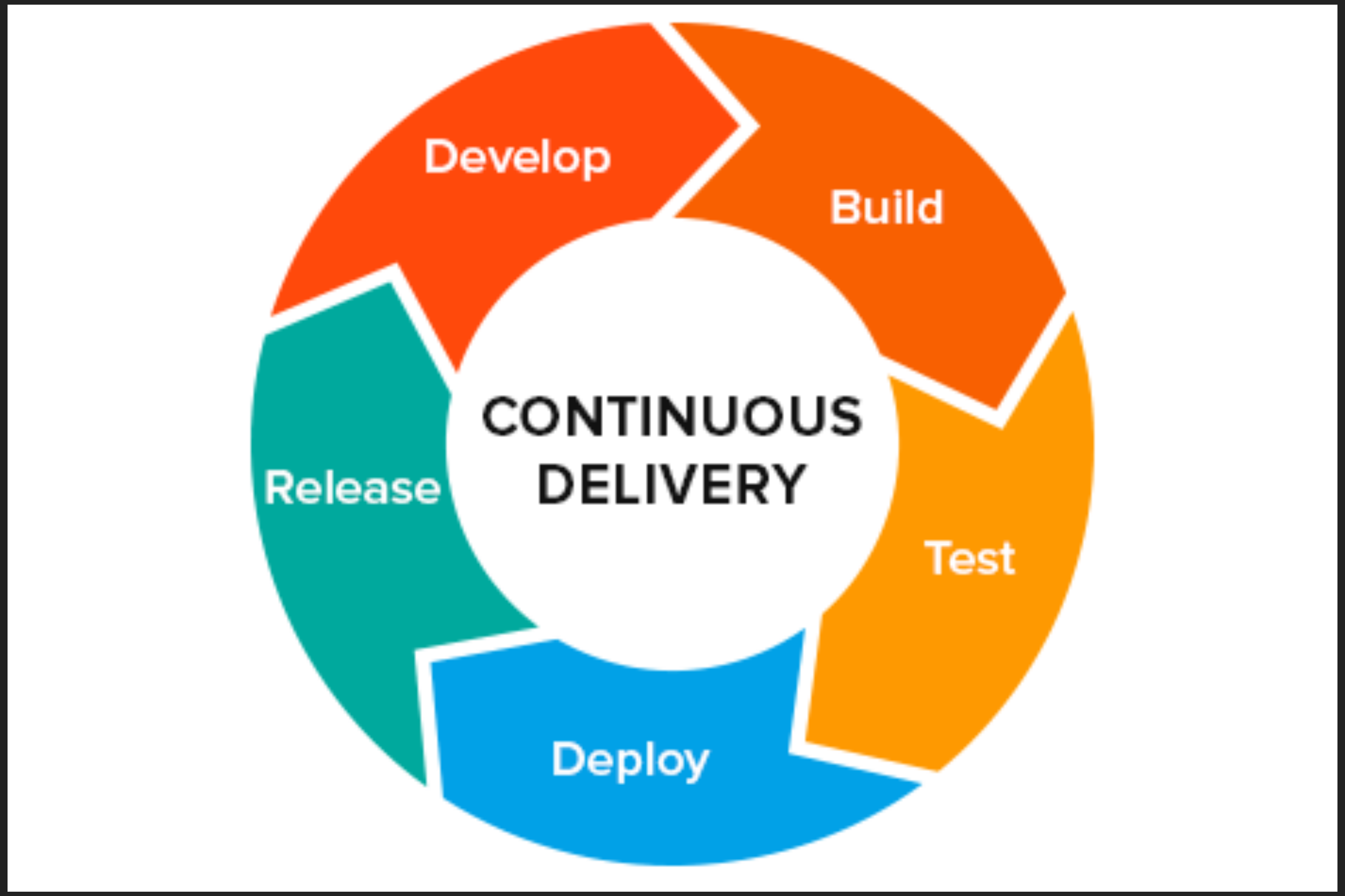
CONTINUOUS INTEGRATION

- The practice of merging all developers' working copies to a shared mainline several times a day



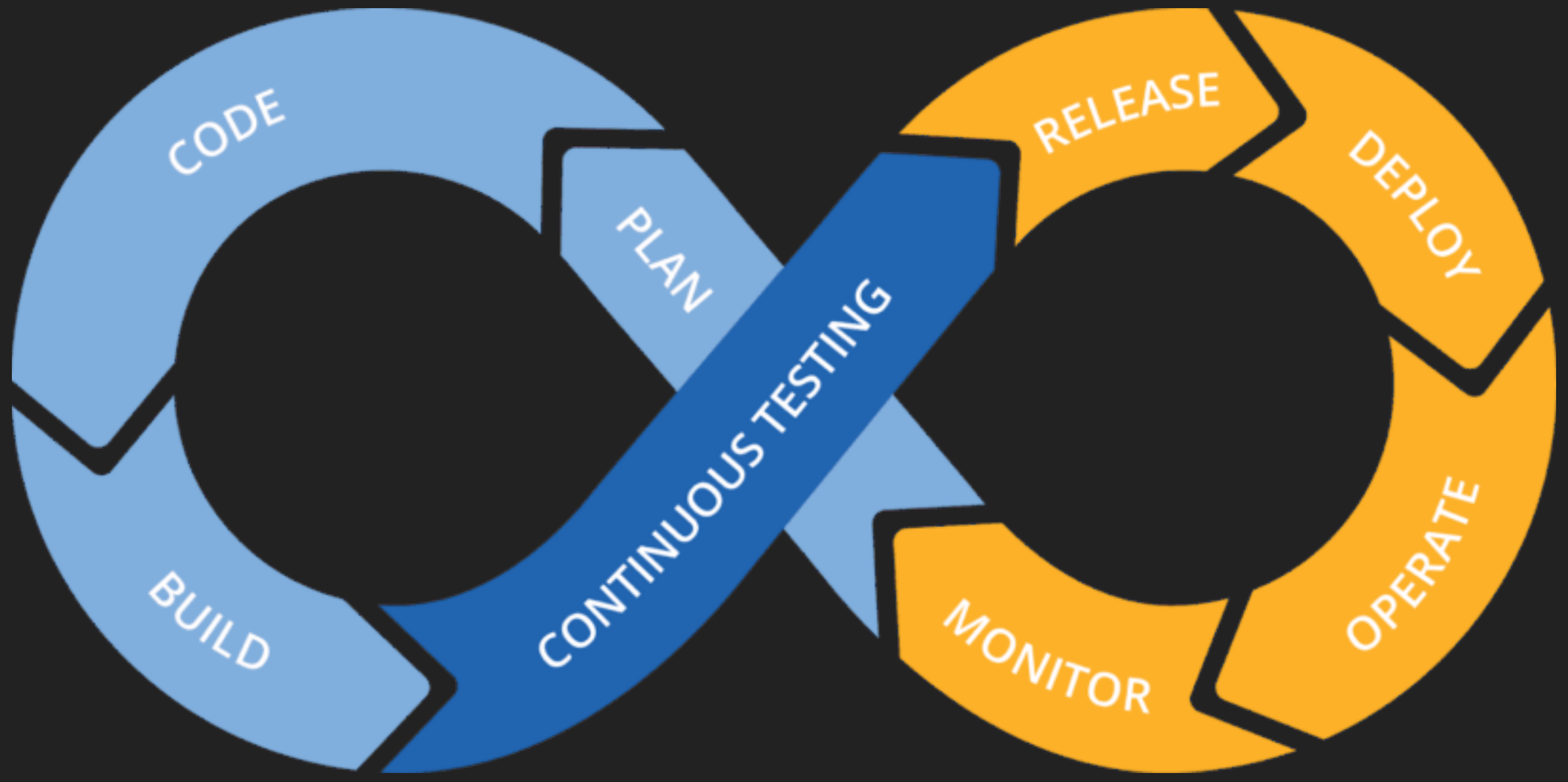
CONTINUOUS DELIVERY

- It is a software engineering approach in which teams produce software in short cycles



CONTINUOUS DEVELOPMENT

- The practice of merging all developers' working copies to a shared mainline several times a day



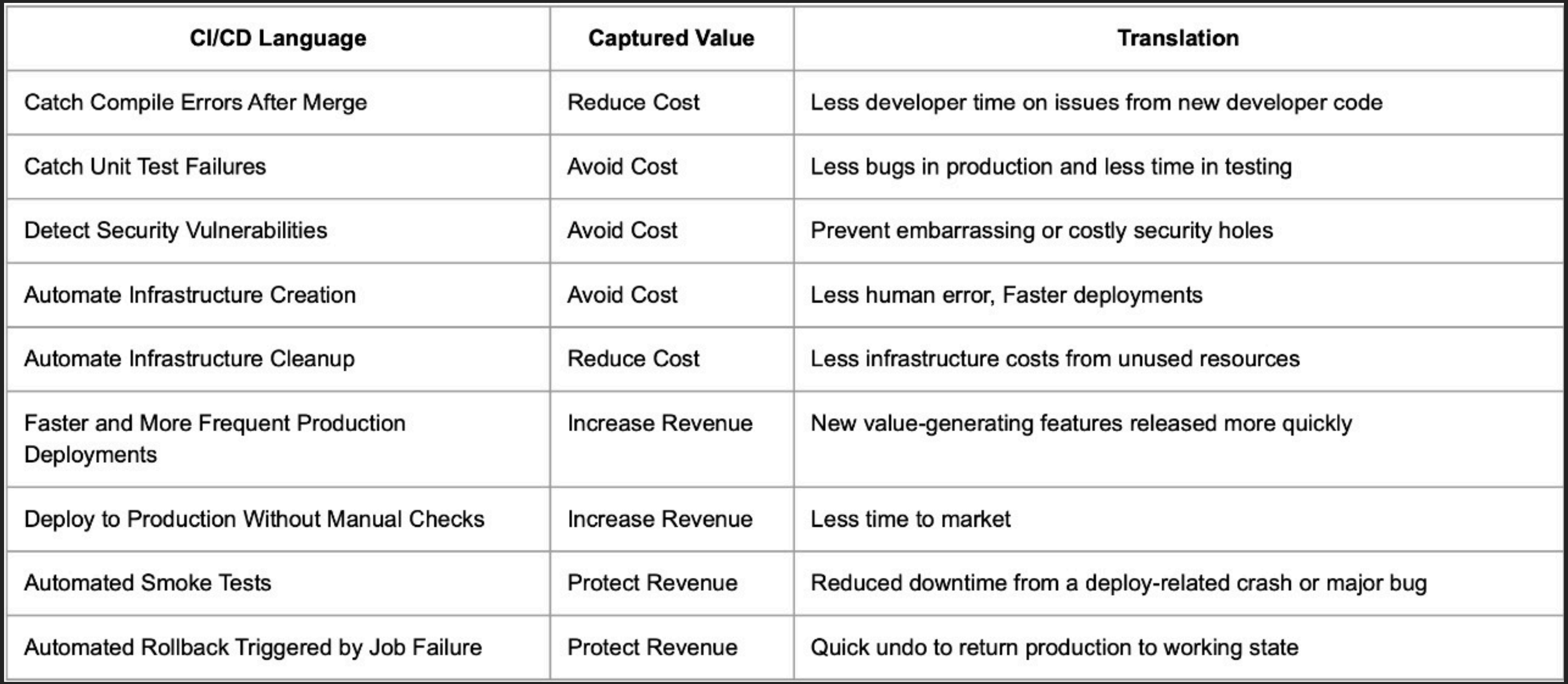
CI/CD FUNDAMENTALS

- DevOps: Set of practices that combines software development and IT operations
- Pipeline: Logical queue filled with the instructions for the computer processor to process in parallel
- Infrastructure as Code: Management of infrastructure using code
- Provisioning: Steps required to manage access to data and resources
- Testing: Steps for ensuring software quality

CI/CD BENEFITS

- Reduce Risk: Less bugs and less testing
- Faster Delivery: Teams can build, test and deploy features automatically with almost no manual intervention using various tools
- Extensive logs Generation: Extensive logging information generated in each stage of the development
- Easier Rollbacks: One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly.

CI/CD BENEFITS (CONT.)



CI/CD Language	Captured 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

CI/CD BEST PRACTICES

- Fail Fast: Revealing failures as fast as possible using CI/CD pipeline
- Measure Quality: To check positive impact of your work improvement
- Only Road to Prod: CI/CD must be the only deployment way to prevent inevitable failure
- Max Automation: if it can be automated, automate it!
- Config in code: All configuration code must be versioned and in code