

# CI/CD CONCEPT

---

# CI

The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making"

In a faster, high quality and deployable artifact Include:

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

# CD

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
-

# CI/CD

*" CI/CD is like a choo-choo train for computer software. It is a system that involves all the train engineers, conductors, train cars, tracks, and controls working well together, to ensure the passengers, which are software-change, get to their destination safely and quickly. " - Marc Hornbeek*

Value	Benefit
Reduce Cost	Less developer time on issues from new developer code
Avoid Cost	Less bugs in production and less time in testing
Avoid Cost	Prevent embarrassing or costly security holes
Avoid Cost	Less human error, Faster deployments
Reduce Cost	Less infrastructure costs from unused resources
Increase Revenue	New value-generating features released more quickly
Increase Revenue	Less time to market
Protect Revenue	Reduced downtime from a deploy-related crash or major bug
Protect Revenue	Quick undo to return production to working state

# WHY CI/CD

Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly. Another thing to keep in mind: increasing code quality with automation also increases your ROI.

---