

CI/CD

Scenario

- Jim and Pam have been working hard for weeks to meet the deadline at close of business.
- They both complete their work and proceed to merge their code into the codebase. Neither gets an error.
- Dwight begins testing and notices things aren't working as expected. The test fails.
- Jim and Pam are now behind schedule trying to figure out what went wrong.
- After 2 days of poring over their huge change sets, they finally spot the error and after another day, they are able to correct it.

Continuous Integration

- Pam and Jim could have alleviated their issues by practicing continuous integration
 - **Continuous integration (CI)** is the practice of merging all developers' working copies to a shared mainline several times a day
 - In other words, developers synchronize their work frequently, so they can more easily find and fix defects
- By employing CI practices, Pam and Jim would have been able find the issue before it had a chance to become a real problem
- In addition, CI employs automated testing, significantly reducing Dwight's workload. This allows him to better focus on the quality of his work, and further reduces the chances clients receiving defective software

Continuous Deployment/Continuous Delivery

- Continuous Deployment (CD) is closely related to continuous integration and refers to keeping your application deployable at any point
- When software is continuously integrated, and built, tested, and security scanned, teams are able to get their software in front of real users faster.
- This not only makes validation faster, but it keeps customers happy, as they are able to see more tangible proof of their dollars at work.

The Bottom Line

- When utilizing CI and CD methodologies, teams are able to:
 - Find and fix defects faster, reducing overall development time
 - Reduce the amount of time spent manually testing software
 - Validate their changes with customers and users faster, allowing them to fix issues before they have a chance to go off the rails
 - Deliver new features in a more timely manner