

UDAPEOPLE: THE CASE FOR CI/CD

CI/CD: WHAT IS IT?

CI: CONTINUOUS INTEGRATION

- Continuous Integration is the practice of continually merging developer's work to the shared mainline.
- CI tools allow us automate tasks like performing tests and builds, which can be triggered on every commit to mainline.
- CI allows us to ensure our new features can be delivered with both speed and confidence in reliability.

CI/CD: WHAT IS IT?

CD: CONTINUOUS DEPLOYMENT

- Continuous Deployment is the practice of automating deployments.
- By automating the deployment process, new features can be delivered frequently and quickly.
- CD de-risks the deployment process by replacing a hero developer having to do deployment 'magic' and with a robust automated process, eliminating human error.

PUTTING IT TOGETHER: CI + CD = CONTINUOUS DELIVERY

When we build a CI/CD pipeline, we're automating the integration process for merging code for new features, and the deployment process for getting those new features in front of customers, allowing us to deliver value quickly, regularly, and with confidence that our new features will work seamlessly, enhancing the customer experience.

THAT'S GREAT, BUT WHY BOTHER?

REDUCING COSTS THROUGH DEV TEAM PRODUCTIVITY

- CI helps catch compile errors and unit test failures fast
- Allows devs to integrate new code more easily
- Reduces dev time spent on testing and minimises bugs reaching production
- Reduces costs to the organisation by freeing up dev time

THAT'S GREAT, BUT WHY BOTHER?

SECURITY AND INFRASTRUCTURE:

- CI automates security audits, allowing us to prevent costs from security issues
- CD automates deployment and cleanup of infrastructure, allowing us to maximize efficient use of infrastructure resources and reduce costs

THAT'S GREAT, BUT WHY BOTHER?

INCREASING REVENUE BY DELIVERING NEW FEATURES FASTER

- Automating deployment to production reliably without manual checks allows new features to be deployed quickly, giving us the competitive edge in time to market
- Fast, reliable deployment means more frequent deploys, allowing us to iterate faster and further increase revenue from rapid improvement of new features

THAT'S GREAT, BUT WHY BOTHER?

PROTECTING REVENUE BY PREVENTING FAILURE

- CD smoke tests ensure deployments work as expected and reduce downtime from failed deployments or major bugs not detected in earlier testing
- When smoke tests fail, rollbacks ensure our production environment remains stable and working
- This ensures our revenue streams are protected and minimises reputational damage from production downtime