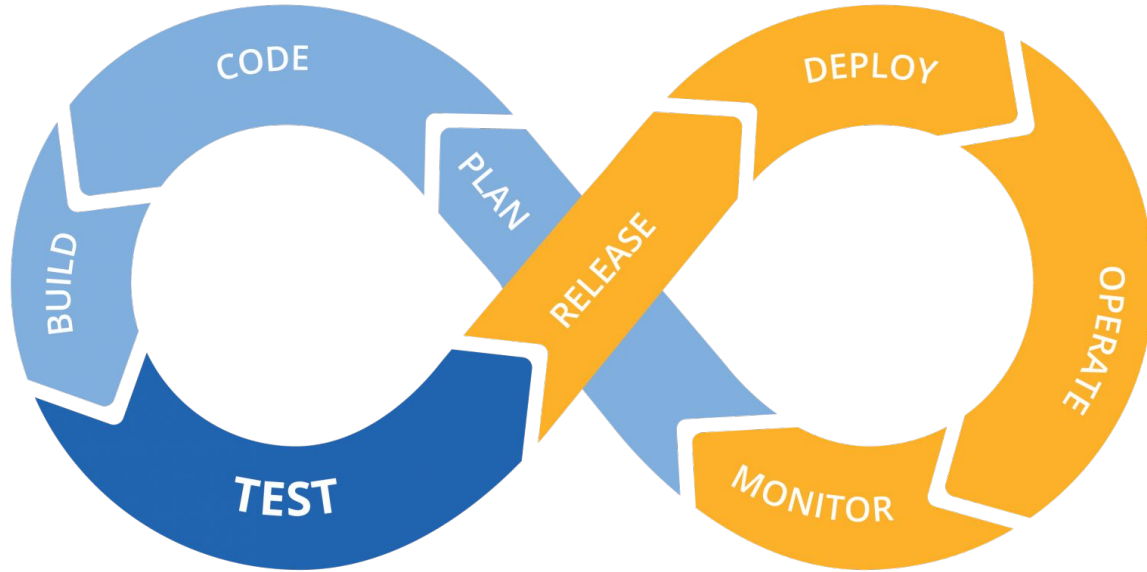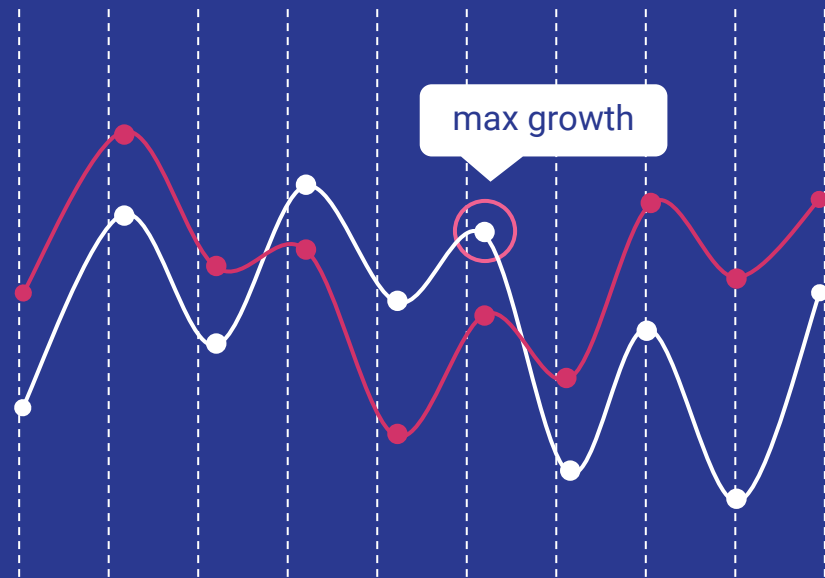# CI / CD Proposal for UdaPeople

By: Miguel Vasquez
Company: UdaPeople
Date: 03 / 11 / 2021

Proposal Outline

- What is CI/CD

- Benefits of Continuous Integration (CI)

- Benefits of Continuous Delivery (CD)

- CI/CD data for Open Source projects

- Disadvantages of CI/CD

- Should we embrace CI/CD?

What is CI/CD?

Continuous integration (CI) and continuous delivery (CD) embody a culture, set of operating principles, and collection of practices that enable application development teams to deliver code changes more frequently and reliably. The implementation is also known as the *CI/CD* pipeline.

CI/CD is one of the best practices for devops teams to implement. It is also an agile methodology best practice, as it enables software development teams to focus on meeting business requirements, code quality, and security because deployment steps are automated.

Continuous integration is a coding philosophy and set of practices that drive development teams to implement small changes and check in code to version control repositories frequently. Because most modern applications require developing code in different platforms and tools, the team needs a mechanism to integrate and validate its changes.

Continuous delivery picks up where continuous integration ends. CD automates the delivery of applications to selected infrastructure environments. Most teams work with multiple environments other than the production, such as development and testing environments, and CD ensures there is an automated way to push code changes to them.
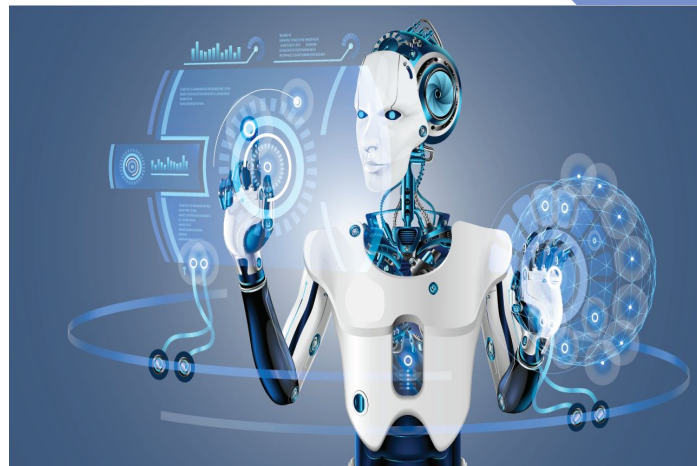
Benefits of Continuous Integration (CI)

Without CI

- Developers merge untested code
- No security checks
- No automatic linting
- Configuration issues between the developers local environment and the staging/prod environment won't be caught.
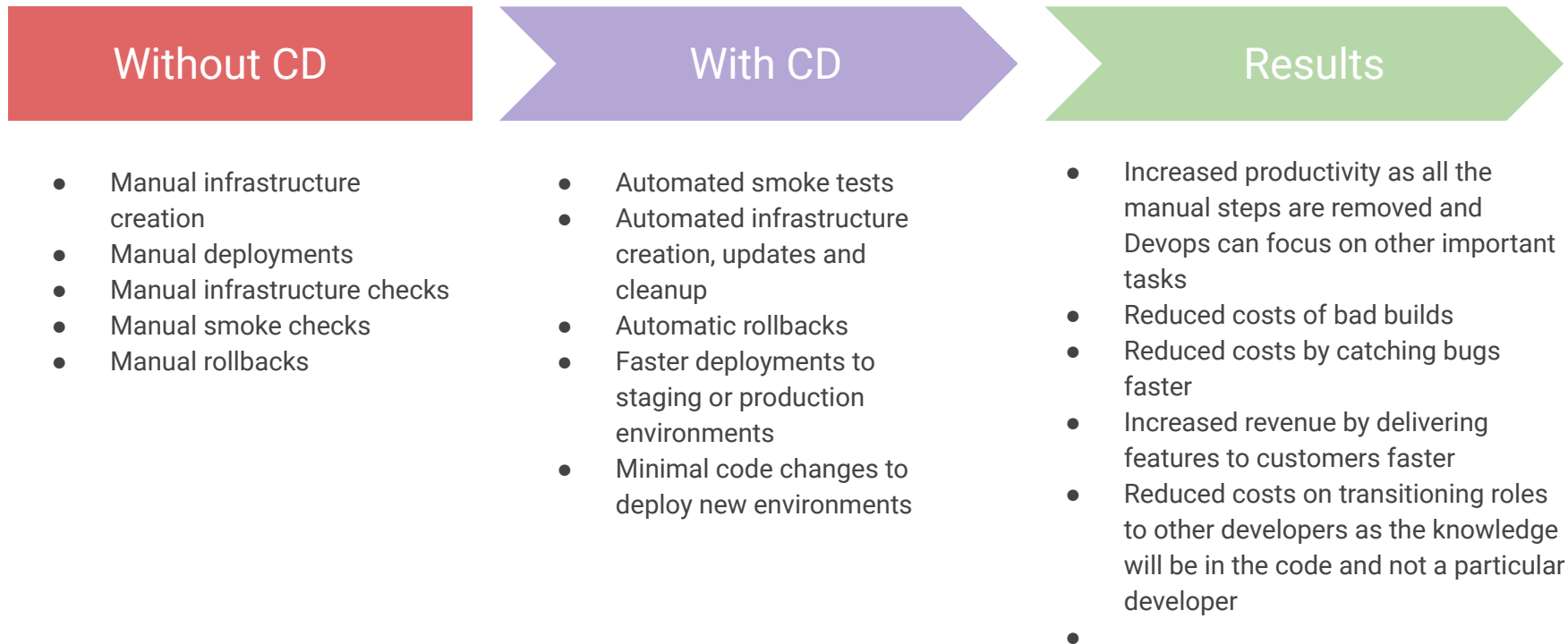
With CI

- Developers will catch unit and integration tests failures
- Static analysis will automatically check for code style, basic security and established best practices
- Bugs are discovered faster with prebuilds

Results

- Less man hours spent on catching issues further down in the pipeline
- Cleaner code translates to clearer code which in turn saves developer time
- Secure code translates to saved cost on audits and any future liabilities
- Higher team morale
- Higher confidence on builds
- Faster time to market

Benefits of Continuous Delivery (CD)

Without CD

- Manual infrastructure creation
- Manual deployments
- Manual infrastructure checks
- Manual smoke checks
- Manual rollbacks

With CD

- Automated smoke tests
- Automated infrastructure creation, updates and cleanup
- Automatic rollbacks
- Faster deployments to staging or production environments
- Minimal code changes to deploy new environments

Results

- Increased productivity as all the manual steps are removed and Devops can focus on other important tasks
- Reduced costs of bad builds
- Reduced costs by catching bugs faster
- Increased revenue by delivering features to customers faster
- Reduced costs on transitioning roles to other developers as the knowledge will be in the code and not a particular developer
-

Research data CI/CD for Open Source Projects

2017 Study of 35k Github open source projects. 1.5M analyzed builds and 442 devs surveyed.
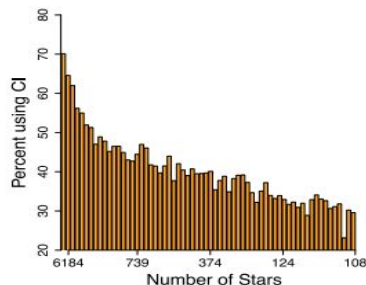
Figure 1: CI usage of projects in GitHub. Projects are sorted by popularity (number of stars).
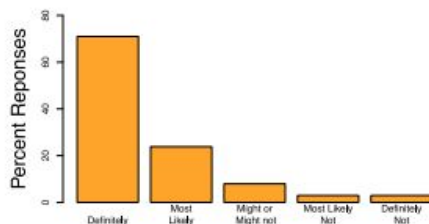
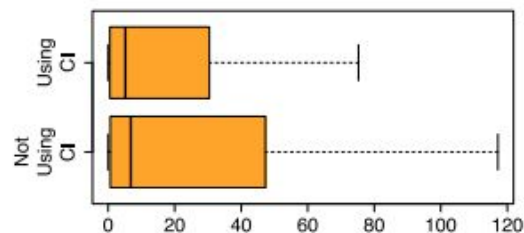Figure 3: Answers to "Will you use CI for your next project?"

Figure 6: Distribution of time to accept pull requests, in hours

Table 4: Reasons developers gave for not using CI

Reason	Percent
The developers on my project are not familiar enough with CI	47.00
Our project doesn't have automated tests	44.12
Our project doesn't commit often enough for CI to be worth it	35.29
Our project doesn't currently use CI, but we would like to in the future	26.47
CI systems have too high maintenance costs (e.g., time, effort, etc.)	20.59
CI takes too long to set up	17.65
CI doesn't bring value because our project already does enough testing	5.88

Disadvantages of CI/CD

Individuals

- High push back from members that are not used to CI/CD practices
- Higher learning curve for non instructed developers

Team

- High degree of discipline and dedication to quality
- Retirement of familiar processes

Company

- Increased initial costs of setup
- Hiring new employees: Devops engineers, QA Engineers and developers.
- Moving away from legacy infrastructure
- Higher infrastructure costs

CI/CD - yay or nay?

Implementing CI/CD will definitely be valuable for us.

The road to implementation will be hard and rocky but we will definitely become a better company as a result.

CI/CD is not a fad there is an increasing trend to implement CI/CD. We have to jump on the train asap!
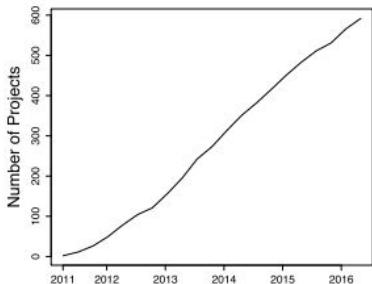

Figure 2: Number of projects using CI over time. Data is tabulated by quarter (3 months) per year.

👍

The advantages of embracing CI/CD are more than staying without it… faster time to market, higher confidence, higher quality and higher team morale are just a few. As well as reduced future costs on time and security.

____