

Proposal for implementing CI/ CD at Udapeople

Continous Integration and Continuous Deployment (CI/CD)

Cardo Eggert (Udacity Cloud Devops Engineer nanodegree)

Continuous Integration and Continuous Deployment (CI/CD) pipeline

- Automated process for building, testing, and deploying software
- Will we lose money because of it? - No, but why?
- Automating building and testing reduces need for manual testing and deployment, which will save us money by reducing costs.
- Automating the deployment process allows us to deliver new features and bug fixes faster to customers, this will make our customers happy and it will protect our revenue and even possible create more of it.
- Automating building and testing also reduces risk of bugs and errors making it to production, which will protect our revenue.

Why should we move away from our current way of deploying?

- We are using a lot of time deploying code manually.
- Think how much more features could be created with that time
- If an engineer spends 8 hours deploying code manually each month, they could use that time instead on more critical topics. (In return creating more revenue)
- The more we grow, the more we have a need for automating the process.
- Less engineers affected by manual testing or deployments = more time for other critical topics
- If 30 engineers were affected by this, we would already save a months worth of engineering time

What is stopping us?

- Need for allocating time for it
- What will take time?
 - Finding tools and technologies that are suitable for us
 - Training engineers to use the CI/CD pipelines correctly
 - Implementing the pipelines for all projects
- This sound like it will cost a lot of money
 - We can get offers from multiple providers and see what fits our needs best
 - It might sound like it but there are more benefits than drawbacks

Conclusion

- Reduce cost - less time wasted on manual testing and deployment
- Revenue protection
 - Automated deployments = more features and faster bug fixes
 - Automated builds and tests = more reliable developments = less bugs to fix
- Increase revenue - more time to spend on new features
- Initial investment seems high but it is worth the risk