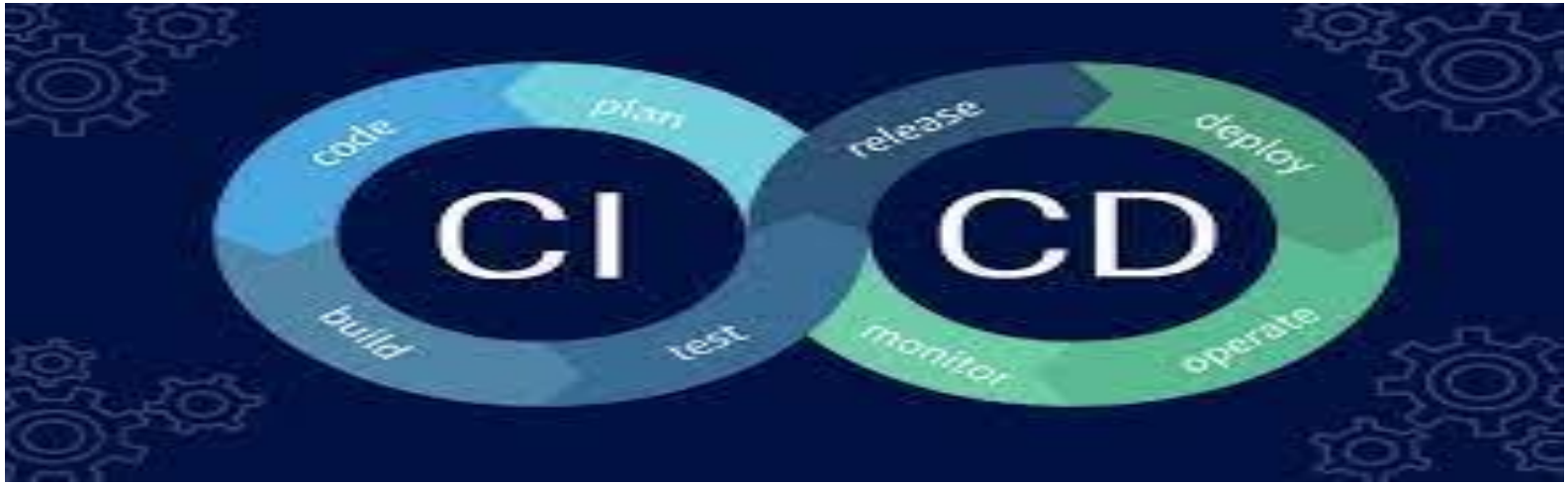
UDAPEOPLE

CI/CD BENEFITS PROPOSAL



OVERVIEW

1- What does CI/CD stand for? The concepts explained
2- What are our current pain points?
3- CI/CD to the rescue. How we could benefit from DevOps principles
4- What are the challenges we will be confronted with?

What does CI/CD stand for? The concepts explained

Continuous Integration/Continuous Deployment (CI/CD) describes the key stages in an automated software development and deployment flow. This flow typically includes design, coding, testing, integration, delivery, validation and phased deployment activities before operation in a target environment.

Since CI/CD attempts to automate the flow from design to deployment, each flow is shaped by the underlying value chain. Feedback from each stage flows back to earlier stages – creating a loop of continuous improvement.

Smaller software changes are key to continuous development and delivery. Rolling out these changes in increments enables large teams to develop code in parallel, while reducing the risk of change, both in development and deployment.

However, a high frequency of small software changes creates an unmanageable amount of manual test, build and deployment activities. To prevent this, CI/CD leverages lean and agile methodologies that rely heavily on automation to enable efficient development and delivery of high-quality software on a continuous basis.

What are our current pain points?

1. Our manual release process is error-prone and always leads to delays of production deployments

2. This in turn often leads to poor software quality since we don't have time for quality analysis anymore

3. Deployments are pretty complex. Only a chosen few experts are able to understand the whole process and tons of hand crafted helper scripts. No smoke tests and rollback mechanisms.

4. We get late feedback from the business department which prevents us from creating flexible solutions

CI/CD to the rescue. How we could benefit from DevOps principles

Problem Statement:

(1) Manual and error-prone release process and

(2) poor software quality

Solutions:

Implement Continuous Integration: automate compiling, testing, code analysis and artifact storage ▸

Automate Infrastructure Creation

Benefits:

1- Cost reduction due to less human errors and faster deployments

2-Reduce complexity and safe manual troubleshooting time

Problem Statement: ▸

(3) Complex deployments and handcrafted automation which often fail. Missing smoke tests and rollback mechanisms.

- **Solutions:**

Automate today's manual deployment steps for smoke tests and rollbacks ▸ Add automated infrastructure provisioning

- **Benefits:**

The truth lies in the source code and not in the heads of one or two experts. This means that regressions and breaking changes in code as well as in infrastructure deployments can be found much quicker and can be resolved for the whole automation process. And as a plus changes are always documented in source code.

Automated Smoke Tests and Rollbacks will protect project revenue due to reduced downtimes from deploy-related crashes and fast and automated rebuilding of production ready state

Problem Statement:

(4) Late customer feedback

Solution:

Implement Continuous Deployment: automated deployment of changes at any given point in time Involve customers and business stakeholders already in deployment process

Benefits:

Faster feedback cycles of customers lead to higher customer satisfaction rates since they are involved right from the beginning of feature development/deployment and not just at a fixed release date

What are the challenges we will be confronted with?

- Establishing CI/CD comes with a high amount of initial cost and learning. At first sight this might seem overwhelming compared to current best practices
- Delivering CI/CD pipelines is not a one time effort, but requires constant support and maintenance as well as continuous development and improvement
- Even though there are some challenges, CI/CD will improve overall business processes and dramatically reduce costs on the long run