UDAPEOPLE

CI/CD Benefits PROPOSAL

OVERVIEW

- What CI/CD stand for?
- What are our current points?
- How we could benefit from DevOpsprinciples?
- What are the challenges we will be confronted with?

WHAT CI/CD STAND FOR?

- CI/CD consist of three major concepts
  - Continuous Integration
    - Continuous Integration describes the process of merging developer branches to the main branch several times a day. CI puts an emphasis on test automation and finally generates a high quality, deployable artifact.
  - Continuous Delivery
    - In addition to Continuous Integration, Continuous Delivery makes sure that changes of a software product can be released quickly to customers in an automated way and at any point in time.
  - Continuous Deployment
    - Continuous Deployment extends Continuous Delivery in such a way that it allows frequent automated deployments without any human interaction. Typical phases in Continuous Deployment are Infrastructure Provisioning, Smoke Testing, Production Deployments, and automated Rollbacks.

WHAT ARE OUR CURRENT 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 complex. Only chosen few experts can 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

HOW WE COULD BENEFIT FROM DEVOPS PRINCIPLES (1/3)

- 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:
    - Smaller Code Changes
    - Fault Isolations and safe manual troubleshooting time

HOW WE COULD BENEFIT FROM DEVOPS PRINCIPLES (2/3)

- Problem Statement:
  - (3) **Easy Maintenance and Updates**

- Solutions:
  - Note within a CI/CD process to perform maintenance during downtime periods, also known as the non-critical hour
  - Don’t take the system down during peak traffic times to update code changes
- Benefits:
  - Make sure the pipeline runs smoothly by incorporating when to make changes and releases. A great way to ensure maintenance doesn’t affect the entire system is to create microservices in your code architecture so that only one area of the system is taken down at one time.

HOW WE COULD BENEFIT FROM DEVOPS PRINCIPLES (3/3)

- Problem Statement:
  - (4) High Cost
- Solution:
  - reduces the number of errors that can take place in the many repetitive steps of CI and CD
- Benefits:
  - Frees up developer time that could be spent on product development as there aren’t as many code changes to fix down

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 are not a onetime 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