CI/CD BENEFITS PROPOSAL

What is CI/CD

- **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 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

BENEFIT Of CI/CD

- Investing **more time** in a release cycle than delivering value
- Going through integration hell every time we finish a feature
- **Code gets lost** because of botched merges
- Unit test suite hasn't been green in ages
- Deployments contribute to **schedule slip**
- Friction between ops and development departments
- **Only one engineer** can deploy a system
- ***Deployments are not cause for celebration***

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

What are the benefits of each practice?

Continuous integration

What you need (cost)

- Your team will need to write automated tests for each new feature, improvement or bug fix.
- You need a continuous integration server that can monitor the main repository and run the tests automatically for every new commits pushed.
- Developers need to merge their changes as often as possible, at least once a day.

What you gain

- Less bugs get shipped to production as regressions are captured early by the automated tests.
- Building the release is easy as all integration issues have been solved early.
- Less context switching as developers are alerted as soon as they break the build and can work on fixing it before they move to another task.
- Testing costs are reduced drastically – your CI server can run hundreds of tests in the matter of seconds.
- Your QA team spends less time testing and can focus on significant improvements to the quality culture.

What are the benefits of each practice?

Continuous delivery

What you need (cost)

- You need a strong foundation in continuous integration and your test suite needs to cover enough of your codebase.
- Deployments need to be automated. The trigger is still manual but once a deployment is started there shouldn't be a need for human intervention.
- Your team will most likely need to embrace feature flags so that incomplete features do not affect customers in production.

What you gain

- The complexity of deploying software has been taken away. Your team doesn't have to spend days preparing for a release anymore.
- You can release more often, thus accelerating the feedback loop with your customers.
- There is much less pressure on decisions for small changes, hence encouraging iterating faster.

What are the benefits of each practice?

Continuous deployment

What you need (cost)

- Your testing culture needs to be at its best. The quality of your test suite will determine the quality of your releases.
- Your documentation process will need to keep up with the pace of deployments.
- Feature flags become an inherent part of the process of releasing significant changes to make sure you can coordinate with other departments (support, marketing, PR...).

What you gain

- You can develop faster as there's no need to pause development for releases. Deployments pipelines are triggered automatically for every change.
- Releases are less risky and easier to fix in case of problem as you deploy small batches of changes.
- Customers see a continuous stream of improvements, and quality increases every day, instead of every month, quarter or year.