Question 1

What do we want to achieve (simplified)?

Accomplish a daily basis releases in no longer than 3 months, to keep the business ahead of the competitor

What's the real challenge to achieve this ambitious goal?

Secure the classique releases (stay efficient for regular tests & validations) and assuring product (business) quality.

What approaches could be considered ?

Let's assume that our QA technical stack is as follows :

- **Automated E2E testing : Cypress.IO (including its dashboard)**
- **Visual regression testing : Percy.IO**
- **Web API testing : Postman / Newman**
- **Performance testing : Apache JMeter**
- **Code versioning and PRs management : GitHub**
- **Mobile Testing (IOS & Android) : BrowserStack**
- **APM solution : AppDynamics**
    - We want to keep our business under control in case of problems and be informed about the potential bugs at the right moment. This could be done through personalized dashboards and deployed AI agents/nodes.
- **CI/CD : Jenkins**
- **Log analytics : Kibana or Sentry ⇒ in order to find the ROOT CAUSE faster**
- **Servers, Infrastructure & System resources monitoring (RAM, CPU, Network etc.) : Grafana**
- **Project tracking and test management : JIRA & Xray**
- **Testing / Staging environment (ISO PROD) : /!\ very important to have /!**

/!\ ONE IMPORTANT POINT TO DISCUSS /!

Should we programe any release on Friday? Cause we're not going to have enough time for a roll-back if needed.

LET’S PLAY !

I’ll start by preparing a checklist and discuss it with my interlocutors in order to confirm it internally at the first place before taking any action (it’s also important to think about a plan B / backup plan).

Why is this checklist mandatory and what is it about?

- To understand the topic deeply and more realistically
- Identify the prerequisites (technically & functionally)
- Define and program a POC (no longer than 3 months in our case) in order to maintain or improve the implementation strategy
- Define a roadmap and create associated JIRA tasks (we can even create an Epic for this project)
- Communicate the beginning of the POC and why we want to do it with team members and managers

I’ll breakdown this project into smaller pieces and here is my strategy :

- **First month :** increase delivery speed by week, meaning, 1 release per week instead of 1 per two weeks **⇒ observe, learn, improve and documentation**
- **First two weeks of the second month:** increase delivery speed by approximately 50%, meaning, 2 releases per week instead of 1 **⇒ observe, learn, improve and documentation**
- **Second two weeks of the second month:** increase delivery speed by approximately 100%, meaning, 4/5 release per week instead of 2 **⇒ observe, learn improve and documentation**
- **Third month :** replacing/migrating the new release strategy with the old one.
    - Approximately 16 releases (if we assume a four-day week) instead of 2

Post-migration actions :

- Communicate the end of the POC as well as the obtained results with team members and managers
- Update project related documentation, release notes etc.
- Upgrade (definitively) Jenkins CI server parameters for this new release strategy and all the headless tests that are supposed to be executed on a daily basis
- Be alerted about the HTTP error response code like 500, 400, 403, 413 etc.
    - This could be done through an customized dashboard
- Sum up the challenge, document best practices and perform a global quality evaluation (technically and functionally)