By Ahmed Hussein

Chapter 1: Chapter one opens with the concept of anti-patterns which are good ideas but terrible implementations after which the book gives us an obvious answer which is Automation of the process of build , deploy, test and release systems. There is emphasis on speed and reliability and improving the work quality by those who work on it. This is reflected in some of the decisions we made on the project to avoid conflicts and redundant design. The main takeaway from this chapter are antipatterns that we should avoid and what we ultimately want to strive for in a software development environment.

Chapter 2: This chapter focuses on the concept of Configuration management which is all about managing the changes to the project and perseriving project artifacts given they might have use. It is all about the control and management that the development team has and how to utilize that to its fullest control to give them an easy path for development and securing that even if a new version of the project has a fatal error that they can backpedal to a previously working version. We have implemented this to ensure we can come back if we commit wrongful updates and work in parallel with each other. The main takeaway is the practices and tools software developers use to maintain steady and secure progress on their code without any risk.

Chapter 3: This chapter is all about continuous integration and its benefits. Continuous integration is in short the idea of continuously pushing code changes into a shared repository to ensure a continuous feedback from automatic tests and easier detection of errors or unintended changes. We have implemented this when after we finish our merges we hotfix issues that occur and see the progress unfold.The main takeaway is to set up and implement an environment that encourages continuous integration to constantly reaffirm steps the development takes to the final product.

Chapter 4: This chapter provides insights into implementing a comprehensive and insightful testing strategy in software development, covering various types of tests, strategies for different project scenarios, and best practices for managing testing activities and defect backlogs. We have implemented this using a Test project and github actions to determine if the code is functioning. The main takeaways are the different types of testing environments the development team needs to ensure their product is up to the quality requirements.

Chapter 14: This chapter is all about Version control and what is referred to as version control system. For our use we use git in this course. Version control is referring to the control and utility that we have of using multiple branches while we work to make work in a large group possible as to not have merge conflicts occur too often. This also plays into the cohesion aspect of grasp where each member gets a branch to work on their part of their work and increase cohesion and make it possible to merge each one of the branches together to get the complete code. Version control is also a tool which we use to go back in time if we want to work from a previous version. We have implemented this using git and creating branches for each one for us to use and creating versions of our code to test out before pushing and implementing into our main branch. The takeaway is the practices we use and the way to increase cohesion and decrease merge conflicts by utilizing the practices of version control.

By Ahmed Hussein

Project Demonstration 1

The purpose of this session is to talk about the project you have developed in two ways:

- project itself, how it works, implementation details, etc;
- how you are managing it with regards to the course content: CI/CD, Project Requirements, testing, etc.

In the first seminar, the focus will be on discussing the fundamentals covered in the first part of the course, including **chapters 1-4** and **14** in the course book, Continuous Delivery:

- Configuration management
- Continuous Integration
- Testing Strategy
- Version Control

Note: All groups are expected to attend the demonstration. Each group have 10 minutes to present their project from a number of perspectives. You do not need to have any slides.

Preparation Instructions

Read the five chapters mentioned above. Each one in the group writes a short reflection (not more than one A4) on how the literature is applicable to your project and what insights they give on how to work in larger projects than this. End the reflection with one sentence per chapter describing the main take-away of that chapter. Post your reflections as comments to the issue created for this seminar. **Post the reflections no later than midnight the day before the seminar.**

Schedule

The demonstrations will happen on Friday, April 26 10:00-12:00 at E1017.

The order of presentations will be given 15 minutes before the session begins. Therefore, it's essential for each group to be well-prepared and adaptable. This will require teams to have their application instances prepared in advance to minimise setup time.