**Version control with Git & Github***Mihajlo Milojević SV57/2023*

* *Introduction*

Hello everyone 👋. I am Mihajlo and today I will be talking about extremely important concept in software development – version control.

* *Table of contents*

But before we begin, I would like to quickly go over the main points of my presentation. Firstly, we’ll talk about what version control is, why is it important and what problems it solves. Then we will see how we can achieve version control using a program called *“Git”* and finally we’ll see what *“Github”* isas well as why it is useful in our profession. So, with no further ado let’s begin.

* *Section 1 – Version control*

What is *“version control”*, and why should you care?

Let’s make a hypothetical project you as a software engineer are working on. This project has lots of features that need to be implemented, so let’s assume you finished some features. You tested them and they work perfectly. Now you move on to implementing a new one. After some coding you test this new feature and find out that it doesn’t work. Now you try debugging your code, and after a while you conclude that not only did you not fix the problem but you also messed up what you had before. You could try just spamming CTR+Z to undo everything or delete all the changes but that would be nearly impossible to the successfully. Maybe you thought in advance and created a backup of your code, but this approach would take a lot of memory depending on size of your project. Wouldn’t it be nice if you can just go back to when it all worked and just continue from there.

That’s where version control come in action. Version or source control is a system that records changes to a file or set of files over time so that you can recall specific versions later. With version control in place, you can make any changes to your project and if it doesn’t work out, just simply roll back to when everything was perfect. Alternatively, you could go thru your changes and see when the problem was introduced into the code. But now you may be wondering how to integrate source control into your project, which leads me to the next part of this presentation.

* *Section 2 – Git*

Git is a program, installed on your machine that does all the hard work for you. When properly configured, git will keep track of all your modification and when you are done, you can just commit it. Committing means that git will take a snapshot, a picture of the current state of your files and add it to its internal database. Once you have something in the database you can revisit it, track the changes thru time or just discard all the changes made after certain commit and affectively restore any previous state of your project.

Let’s go back to our hypothetical project. You have added version control to your project, and now it’s time to share your code with the rest of the team. We need to somehow effectively share our code with everyone and also sync all changes, made by anyone into one final project. How would we do that? That’s why we have Github.

* *Section 3 – Github*

Unlike Git, which is a program that runs locally, Github is a platform that allows developers to share code, work together and integrate seamlessly. All you need to do is create free account and you can start using it. Github also makes all your code portable. You don’t need to carry a flesh drive, or mail your code every time you need it, simply clone your repositories, make the necessary changes and push them back on Github. Now you, or anyone else can simply pull changes and always have the most recent version of your project.

* *Conclusion*

In conclusion, version control is a must have skill any software engineer should possess. I didn’t have enough time today to go in depth and really explain all the details and really show you how to use tools I have talked about. I really recommend you find some courses online to learn version control.

* *Outro*

With this I would end my presentation. If you have any question or want to learn more feel free to ask now or contact me. Thank you for your attention. Goodbye