



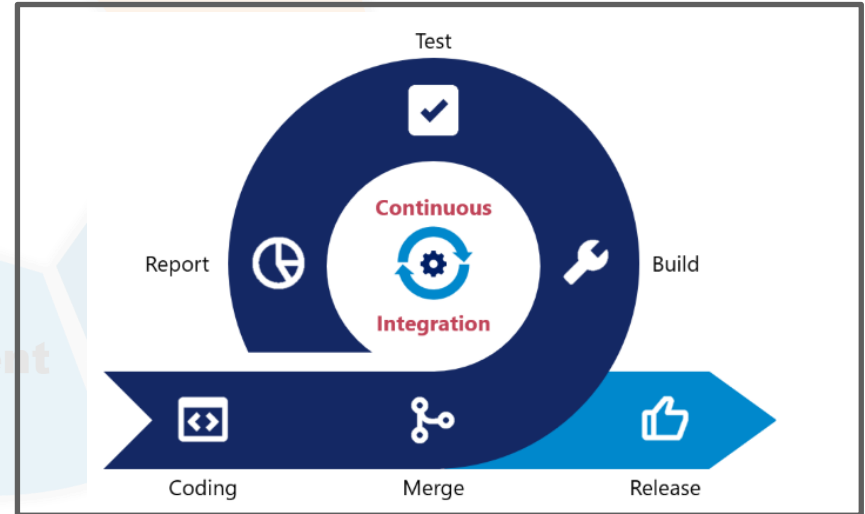
**Name:** Máté Losonczi, BSc student, 8th semester

**Project type:** thesis project

**Topic:** Automation of build and test processes with Jenkins pipeline

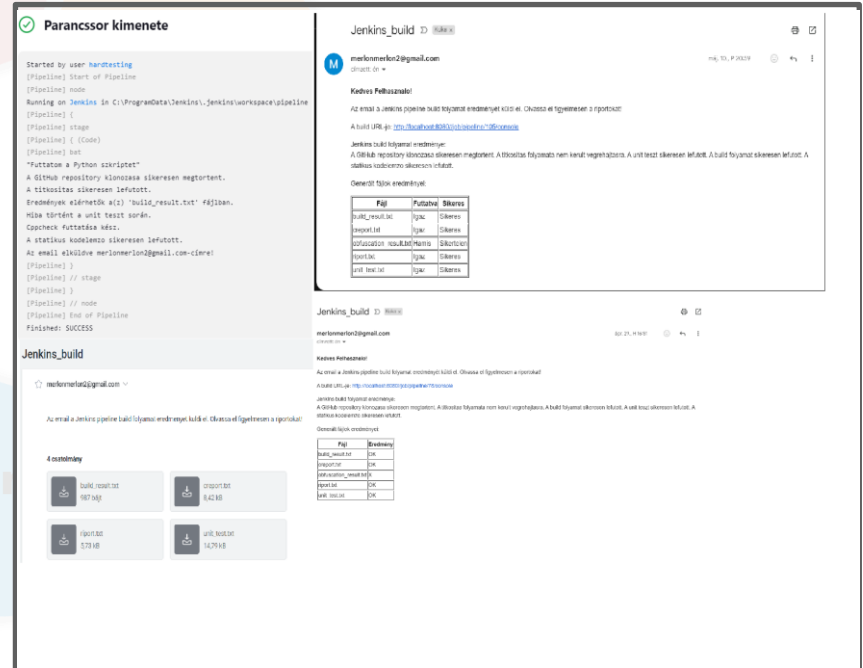
**Supervisors:** Dr. Szilveszter Pletl

- During automotive software development, the automation of integration steps greatly increases reliability and saves developers from significant manual work.
- Both factors are desirable, which is why the practice of so-called continuous integration is spreading, during which the completed work packages are integrated and tested daily.
- The goal is to create a script in Python that can be used to implement automated build and test processes.
- Continuous integration itself is a DevOps software development practice where developers regularly merge code changes into a central repository, after which automated builds and tests run.



# Experiments

- The process was already given, since we did not invent continuous integration.
- Our task was to create a script that downloads the latest version from the version tracker, optionally encrypts certain parts of the code, builds, runs the unit test and performs static code analysis. These are all run by the Jenkins job based on parameters.
- It also sends an email about the results of the executed processes to the email address obtained from the Jenkins parameter, which includes the name of the process, whether it was run and the success of the output. It also sends the URL of the console for the given run.
- There was a lot of trial and error while creating the script and so many possibilities for error. I tried to handle possible errors in the appropriate way, to test as many scenarios as possible.



The image shows two screenshots related to a Jenkins build. The left screenshot is the Jenkins console output for a job named 'Parancssor kimenete'. It shows the build starting at 10:22:08, downloading code, running a Python script, and performing static code analysis. The build finished successfully at 10:22:10. Below the console output, there are four download links for artifacts: 'build\_result.txt' (187 kB), 'report.txt' (4.42 kB), 'report.txt' (578 kB), and 'jUnit\_Results' (14,794 kB).

The right screenshot is an email notification titled 'Jenkins\_build' sent to merlonerion2@gmail.com. The email contains the same console output as the left screenshot, along with a table summarizing the build results:

Fájl	Futatója	Sikeres
build_result.txt	True	Sikeres
report.txt	True	Sikeres
report.txt	True	Sikeres
jUnit_Results	True	Sikeres
jUnit_Results	True	Sikeres

Below the table, there is another section titled 'Jenkins\_build' with a similar table:

Fájl	Erősevény
build_result.txt	OK
report.txt	OK
report.txt	OK
jUnit_Results	OK
jUnit_Results	OK

# Results & future work

- The program also works by specifying parameters on the Jenkins interface. The truth value of a parameter determines the output of the program run.
- If the user marks everything as true, the program synchronizes the project from the version tracker, encrypts the 'src' folder, performs the build process with the encrypted files, then runs the unit test and then the static code analysis. It then notifies the user by email.
- The program certifies successful execution for all possible outputs. Cloning affects the whole process, if the project exists in the folder and is not overwritten with the latest version, it will not clone the project, but it will perform all other processes.

## Future work:

- Share more information with the user in the email, possible details.
- Contact the testers to see what kind of testing process they could add to this.

The screenshot displays the Jenkins build interface for a job named 'Jenkins\_build'. The 'Parameters' section shows several boolean parameters, all of which are checked. The 'Build' section shows the build log, which includes the following table:

Folyamat	Futtatva	Eredmény
Clone	Igen	Sikeres futás felírás nélkül
Build	Igen	Sikeres futást
Obfuscator	Igen	Sikeres futást
StaticCodeAnalyzer	Igen	Sikeres futást
UnitTest	Igen	Sikeres futást

The email notification on the right shows the subject 'Jenkins\_build' and the body text: 'Kedves Felhasználó! Az email a Jenkins pipeline build folyamat eredményét küldi el. Olvassa el figyelmesen a riportokat! A build URL-je: <http://localhost:8080/job/pipeline/200/console>'.