Version Control Systems, Build Automation and Continuous Integration -- Exercises

# Systems Engineering Laboratory 1

**Note:** Please treat these exercises also as professional work. For example, instead of *asdfg* use more meaningful commit messages like *Add acceleration feature* or *Fix #5* (you can find detailed advice in the [How to Write a Git Commit Message](https://chris.beams.io/posts/git-commit/) and the [Git Style Guide](https://github.com/agis/git-style-guide) posts).

# Initialization

1. Check on the GitHub website your base repositories initialized by the course administrator
2. Initialize your development environment (start the proposed virtual machine and overview the tooling)

# Markdown exercises

1. [local] Clone your project using the local git tool into your development environment.
2. [local] Edit and add some notes to the Readme using the Markdown syntax. Present at least three different formatting in your new content. (Use your development environment to do the changes instead of the Web UI)
3. [local] Commit, push your changes and check the results on the Web UI.

# GitHub Flow Exercises

1. [web] Define a new feature (or change request) on your project and create a GitHub Issue Ticket to maintain it. (You can "request" a bugfix or a new feature)
2. [web] Create a branch for your issue.
3. [local] Pull your project and change to the new branch.
4. [local] Do a local build using Gradle. Test the project by executing it.
5. [local] Do the changes on your code according to the defined ticket. (You can do changes, build and test the project on your development environment)
6. [local] Commit and push your changes to the repository.
7. [web] Create a pull request (merge request).
8. [web] Check your pull request, check the changes, comment it and if it is ok, then approve and merge it.
9. [local] Check the git log command to show the change history.

# Merge conflict

1. Figure out how to generate a merge conflict.
2. [local] Create two new branches (branch-A, branch-B).
3. [local] Checkout branch-A, edit one line in a file and commit the changes.
4. [local] Checkout branch-B, edit the same line in the same file on a different way and commit the changes.
5. [local] Switch back to the master branch and merge branch-A and branch-B afterwards. Check the results and if a merge conflict occurs, then resolve it.

# Circle CI

1. [web] [Login](https://circleci.com/) to Circle CI using your existing GitHub account.
2. [web] Set up your repository. It can be found under the ‘Add projects’ tab.
3. [local] Follow the instructions:
   1. Add and edit .circleci/config.yml to your project.

Hint:

version: 2.1

jobs:

build:

working\_directory: ~/<<repository\_name>>

docker:

- image: circleci/openjdk:8-jdk

steps:

- checkout

- run: ./gradlew check

* 1. Commit and push your project in order to trigger a new build.

1. [web] Check the web, check your build, console output and result. Check the results of the JUnit tests.
2. [local and web] Do some changes to generate build error. Commit, push and check the results on the web.
3. [local] Fix build error and add one new JUnit tests to your project.
4. [local and web] Commit and push to trigger build with JUnit tests and check the results.

(Megjegyzés: Lehetne alapértelmezetten 2-3 test a repoban úgy, hogy az egyik direkt elszálljon (teszt jó, implmentációs hiba a kódban). Ezek után a feladat leírás így módosulna:)

1. [web] Check the web, check your build, console output and result. Check the results of the JUnit tests.

Hint: Yes, one of your test cases will fail. Do not panic.

1. [local] If any of your test cases failed, fix it and add one new JUnit test to your project.
2. [local and web] Commit and push to trigger build with JUnit tests and check the results.

(Megjegyzés: Korábban itt az ‘Add external dependencies’ feladat volt. Ezt átraktuk a 3. laborhoz, amikor a sonarcloudot kell integrálniuk)

# Add external dependencies

TODO, lecserélni más feladatra

# Finishing the lab

1. Commit and push everything to the repository, check if the build is success and present it to the teacher.