

## Lab Exercise: Setting up Jenkins and GitHub to cooperate on Continuous Integration

In this exercise you will setup Continuous Integration on our Jenkins server, with build and test of the Calculator-exercise you started last week.

The Jenkins Project will pull the latest version from your GitHub repository, before it builds and tests this latest version, when it is triggered manually or automatically.

You will also setup GitHub to trigger your job on the Jenkins server, when new file versions are pushed to your repository on GitHub.

**Remember!** Whenever you have done a change (implemented a new test or a working part of a method), commit your work with a decent commit comment. When you have finished a suitable amount of work, do a Pull – Test – Push cycle, as explained in the lecture slides.

1. Prepare your project for using dotnet and Jenkins as test runner instead of Resharper: Use the NuGet manager to install the **NunitXml.TestLogger** and the **NUnit3TestAdapter** packages on your test project. Commit with comment and push to GitHub.
2. Each team Creates an account on our Jenkins server <http://ci3.ase.au.dk:8080>. Use a name, identifying your team and a password you agree on.
3. Create together a Jenkins New Item – called a project. Use the already defined project **TemplateCoreBuildTest** to copy from, and fill in your repository and other relevant information, as indicated in the template project – e.g. the correct name of the solution and projects, test result file, etc., where it is necessary. Look for a pair of \*\*\* in the template. The \*\*\* indicates that something must be replaced with the correct information instead of the \*\*\* info \*\*\* sequence.
4. Use the Build Now option to verify, that the project can obtain the code from the repository, build and run the tests. Correct until this is the case.
5. Sign in and go to your repository on GitHub.com. Select Settings for the repository (only the owner can do that, not collaborators).
6. Select the Webhooks section.
7. Add a webhook, using the URL: <http://ci3.ase.au.dk:8080/github-webhook/> Don't change the default content type. Select "Just the push event". Click the Add Webhook button.
8. Now, one from your team shall create a change in the project. Build and Test. Commit the change to the local repository. Finally do a sync: Pull – Push to your remote GitHub repository.
9. Goto the Jenkins server. After a few seconds (10-15), a new Build for your Jenkins project should start automatically, if you have set up your webhook correctly. If not, check what went wrong, looking at the GitHub server.
10. Continue working in parallel on the Calculator, syncing when you have a working change with tests. Watch the Jenkins project as it tracks your growing number of test cases, etc.