FH OBERÖSTERREICH

Exercise 2

Rules

- Individual work
- Create a public repository (GIT/SVN)
  - Provide a readme.pdf including the URL
- Use the following folder structure
  - dev: contains a project
  - doc: contains documentation, bug reports, meeting protocol, protocol

- **Readme Example:** ex02_mcXXXXX.pdf

Grading

- Protocol
- SVN History

Step 1

Version Control

- Use an existing JAVA project (e.g. excercise, semester project)
- Put this project under version control from inside an IDE plugin (not Tortoise)
  - Eclipse: http://subclipse.tigris.org/
- Either use GIT or SVN
- Only check in relevant files (no temporary files, binaries, etc)
  - Correct use of svn ignore
- Relevant for grading
  - What is in the repository at the intial check in?
  - Have there been changes and commits after the first chek in?
  - Has it been used over a longer period of time?
  - Does the project build?
- Questions (Protocol)
  - Do some research about the benefits a distributed version control system, in contrast to a central VCS, could bring

Step 2

Bugtracking

- Introduce a bug to your application
  - Wrong calculations, missing functionality, etc.
  - Exception on action, etc.
- Commit the application
- Create a bug tracking account
  - Use bug tracking system to report bug
    - http://lighthouseapp.com/, http://www.jetbrains.com/youtrack/index.jsp, http://www.atlassian.com/software/jira/overview
    - Export bug report to PDF or provide screenshots
  - Solve the bug in the code and commit
  - Complete the bug report
    - Export bug report incl. case history
    - Put the final report to the doc folder in your repository

- Question
  - What are the main points a bug report consists of?
  - What could be the reason why a bug report is sent to the creator, before a case is closed?
  - Describe at least three different bugtracking system and find out for which projects they are used

Step 3

Wiki

- Use the Wiki from your bugtracking system or create a new one at any host
- Create a meeting protocol based on a Wiki article
  - Could be either a fictive meeting or a project meeting
  - Be sure to include all relevant points for a protocol
- Add the protocol to the doc folder in your repository

Question

- Describe at least three different use cases where you would use a wiki during your semester project.

Step 4

Ant (Java only)

- Create a build script for your project that allows building your application within one step
  - Your script should clean and build your project
  - It should automatically execute any unit tests included in the project and halt on failures
  - If the unit tests run without error, it should pack the project to a zib and automatically create the Javadoc documentation
- Add the build script as well as the output your script produces to the doc folder
- Document your workflow