

TDDE46: Software Quality

Seminar 4

Theme: Automatization

Objectives:

1. To understand software quality automatization tools
2. Evaluating bachelor projects
3. A presentation about a topic of interest
4. A discussion about the challenges/issues/problems, faced by TDDE46 students in coaching

Seminar preparation:

1. All students must have made logbook entries. Prepare a five-minute presentation about one of them. Make sure that each of you have a unique study. To coordinate this, we have an excel file called Presentation at Seminar.xlsx in the course documents folder on Lisam. Enter a reference or a one-sentence description of what you will bring to the table. If someone else already entered the same material, take your second-best choice. If you have found a topic that is highly interesting for you, make a presentation about that. You don't need to stick to the theme of the seminar.

https://liuonline.sharepoint.com/:x:/r/sites/Lisam_TDDE46_2022VT_O2/CourseDocuments/Presentation%20at%20Seminar4.xlsx?d=we30bec6b421440479989a96a774ab5eb&csf=1&web=1&e=0hZoMI

2. Group 3 makes a little longer power-point presentation on an interesting topic. Each topic should cover 3-4 peer-reviewed papers. The topic does not need to be connected to the seminar theme. The requirements on presentation techniques and critical discussion are high.

During the seminar:

10:15-10:35 (group) Individual presentation of your logbook entry within your own group focusing on:

1. What are the main findings from your study?
2. What problems, challenges, advantages, solutions, etc. have been shown in the study?
3. Mention two things you learned? Take notes about this!

10:35-11:00 (cross-group) Discussion & Brainstorm

In the new constellations, present things learned in your homegroup. Discuss one or several of the following:

1. Do the result from the presentations trigger further topics that would be interesting to study? What would you like to know more about?
2. Can the outcome of the presentations be used in the coaching project?

3. What might be typical processes of the bachelor projects?
4. How can these be measured and improved with reasonable resources?

11:00-11:15 Break

11:15-11:35 (all) Presentation of a chosen topic by a group Try to answer the following questions:

- 1) Why does a selected topic require attention from researchers and practitioners?
- 2) Does the problem/issue/challenge relate to your interests?
- 3) What is a suitable method, presented in paper or in your opinion, to address the problem/challenge?
- 4) What solutions are presented in paper or in your opinion, to address the problem/challenge?
- 5) How can we use our knowledge from this presentation/reading towards improving the BS projects? (Important)
- 6) How can we use our knowledge from this presentation/reading towards improving our methods/ways for helping BS students? (Important)

11:35-12:00 (all) Discussion about plans for the bachelor projects

Topics can include Coaching style, communication means, how to evaluate the projects.

Out of ideas about what to study?

Automatization of software quality is found in several (sometimes overlapping) categories

- Regression testing tools, such as: Junit, pytest, cppunit,
- Automated build tools, such as: Jenkins, Maven, Travis, Tekton
- GUI testing, such as Selenium
- Capturing software metrics, such as Eclipse Metric Tool, SonarQube, Visual Studio
- Analyzing and visualization of data, R, QlikView, Kibana, Grafana
- Static analysis of code, such as Clang, Cpplint, SonarQube, Visual Studio
- Profiling, such as MS Visual Studio Profiler, J Profiler
- Performance measurement, dynamic analysis, such as k6 Cloud, Valgrind
- Repos, such as Git, Bitbucket
- Defect databases: Bugzilla, Jira
- Support tools in processes, such as Asana, Open project, IJI Project library, Excel (sic!)

Many tools do not necessarily have software quality as their main goal but can contribute to quality due to their usage. For instance, testing is used to find bugs, but with statistical data and clever selection of test cases you can build a model predicting the reliability based on the outcome from testing.