STaR 2023 Group project

The group project for STaR is done in groups of 5 students, formed by the students themselves. The aim of the project is twofold: to apply the techniques discussed in the lectures to actual software, and to explore a risk assessment or software testing topic not covered in the lectures. The group project consists of two parts:

1. Group project assignments:

The group project assignments are due each week. In these assignments, each group applies the theory of the lecture to a code base. The code base must be supplied by the students themselves: for example it can be a project from an earlier course, open source software, or software from your job. In the latter case, your employer has to agree, and the code must be accessible to all group members. The code can be about anything, but it should satisfy the following requirements:

- You must be able to execute unit tests
- The code must be large enough (at least +/- 1000 lines)
- There must be several modules.

If you have trouble finding appropriate code, please contact us.

2. Group project report:

Each group also has to do a larger project, of which they have to write a 4-page report, to be handed in April 18. This project also has to be presented on March 28. Such projects can be of one of the following four forms:

1. Apply a risk assessment or software testing method not covered by the course to the code base mentioned above. Some examples of risk assessment methods:
    a. Agent-based risk modeling
    b. Software Fault Trees
    c. Bowtie diagrams
    d. STAMP
    e. something else (please confirm with us)

    Some examples of software testing methods include:
    f. White-box, coverage: based: Emma, Junit
    g. Quality metrics: Sonar Cube
    h. Behavior-driven testing: SpecFlow, CuCumber, Karate, Cathlon
    i. Concolic testing: SAGE, DART
    j. Testing of web services: REST, Selenium
    k. Test frameworks: Gerking, RObot
    l. Model-Based testing: JtorX, MatLab, Motbat
    (see http://mit.bme.hu/~micskeiz/pages/modelbased_testing.html)
    m. Fuzzing tools: SAGE

n. Java testing: Diff Blue
o. Use big data analysis tools (e.g. Weka) in testing, e.g. clustering test cases.
p. something else (please confirm with us)

2. Do a small research project on the theory of risk assessment models. Some example topics include:
    a. *Transform fault trees into BDDs*: As explained in Lecture 2, fault trees analysis techniques first encode the tree into a compact representation, called a Binary Decision Diagram. The goal of this project is to implement this encoding. Since the efficiency of the encoding depends on the variable ordering (i.e., an order on the parameters of the Boolean formula), an important question is which variable work best. Especially: can you derive heuristics to find good variable orderings?
    b. *How good is your fault tree?:* A problem that risk analysis face in practice is that they like to see how good their fault tree actually is. Are there rules of heuristics that can tell you if a fault tree looks reasonable?
    c. *Model metrics:* For source code, there are a lot of metrics that tell you how complex the code is: cyclotomic complexity, the coupling and coherence of classes, etc. For transition systems, such methods are rare. Thus, the objective of this project is to develop complexity metrics for other models. Such models can be
        i. Finite automata / transition systems
        ii. Fault tress
        iii. UML models
        iv. ... other models you like.

        Obviously, the number of states and transitions are examples of such metrics. The question is, however, if there are more interesting and relevant metrics.
    d. *Tutorial on risk assessment methods:* take one of the methods of 1a—d and create a tutorial for it: Explain the how the model works (modeling constructs, principles) and what you can do with it (analysis types). Include various examples to illustrate the model and their constructs, and analysis types.
3. Risk assessment case study: Do an extensive risk assessment for a problem that you like. Choose a formalism of your choice to model the risks (for example, a method from the lectures, or 1a—d). Come up with specific measures and recommendations. You can only make a useful risk assessment if you have sufficient information. A good source for risks is the Dutch Safety Board. All their investigations are publicly available at https://www.onderzoeksraad.nl/en.
4. Bring your own project: if there is another topic you are enthousiastic about and want to explore in more depth, please discuss with us whether it is appropriate for this course's group project.

The final report should be 4 pages excluding appendices, references, and pictures, and should follow the form below:

1. Introduction
    - What problem did you solve?
    - Why is that problem relevant?
    - How did you solve it? (sneak peek, details in 3)
    - What are the results? (sneak peek, details in 4)
2. Background
    - Introduce the main concepts you are using
3. Methods/Approach
    - How did you solve your problem?
4. Results
    - What are the results of your method?
    - What worked well, what did not work well, and why?
    - Tables are good, graphs even better
5. Conclusion
    - What is the main conclusion from your approach?
    - What are interesting topics for future research?
6. References
    - Unlike BSc/MSc thesis you do not need an extensive state-of-the-art, and a few references will probably suffice.
7. Appendices
    - e.g. raw data underlying your graph
    - We will only read it if we feel like it, so make sure your full story is told in the main paper