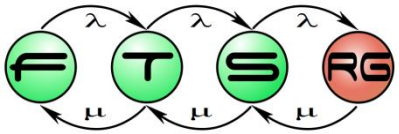
Soft**w**are and **S**ystems **V**erification
(Szoftver- és rendszerellenőrzés)

http://inf.mit.bme.hu/en/edu/courses/swsv

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Introduction

Zoltan Micskei (lead instructor)

Istvan Majzik (instructor)

Akos Hajdu (instructor, labs)

Kristof Marussy (labs, assignment)

teaching assistants (lab, assignment)

Structure of the course (2/1/0/v)

- Lecture
  - Presenting problems and methods + exercises
- Lab sessions
  - Trying out tools on smaller exercises
- Home assignment
  - Applying the techniques on a (complex) application
- Exam
  - Summarizing knowledge

Structure of the course (2/1/0/v)

- Lecture: Wednesday 10:15-12:00
- Lab: Thursday 14-16 or 16-18 (every other week)
  - Table on website for course codes (G1, G2, GA)
  - Days off: 12 Sept. (Sport day), 23 Oct. (Holiday)
- Home assignment
  - Teams with 4 members, using GitHub
  - 3 phases (4rd, 6th, 9th week)
  - See the web page for details
- Exam
  - Written exam: theoretical and practical part

News and information

https://inf.mit.bme.hu/en/edu/courses/swsv/news

- **News only here**
  - No separate Neptun messages
- **Use RSS** 
  - (→ use IFTTT for publishing to other channels)

Requirements

- Lab
  - “Absence should not exceed 30%” -> 1 lab can be missed
  - Assessment: screenshots in the GitHub repository
- Home assignment
  - 0-10 points for every phase, needs at least 4
  - Every phase has to be completed
  - Retake/delays: no retake, no late submission
- Signature
  - (Lab OK) AND (Home assignment OK)
- Final grade
  - 0.5 * Home assignment + 0.5 * Exam

Learning outcomes (LO)

- 3-4 LO assigned to every lecture
  - What are the expected outcomes
  - Helps for exam preparation

- Knowledge level
  - **K1 Remember** (define, list, recognize, identify…)
  - **K2 Understand** (summarize, classify, describe…)
  - **K3 Apply** (use, perform, apply…)
  - **K4 Analyze** (evaluate, assess, integrate, select…)

Details: Declan Kennedy. „Writing and Using Learning Outcomes - A Practical Guide”, 2007

Home assignment

- **Goal**
  - Like working professionally on realistic tasks
- **Expectation** (Master level subject)
  - Not only trivial/practiced tasks
  - More than one possible solution
  - Think, analyze…
  - Ask questions (in breaks, labs, Q2A)
  - Own decisions and arguments
  - 32 hours of work per team member (!)
- **Assessment**: quality and not quantity

Home assignment (team registration)

- Form: see the news next week
- Finding team members:
  - LAB0 sessions
- LAB0 will introduce to the infrastructure, tools

Example home assignments – GOOD

Links to the artefacts for evaluation:

SRS

- SRS Inspection Aspects & Review Process
- Commented SRS document

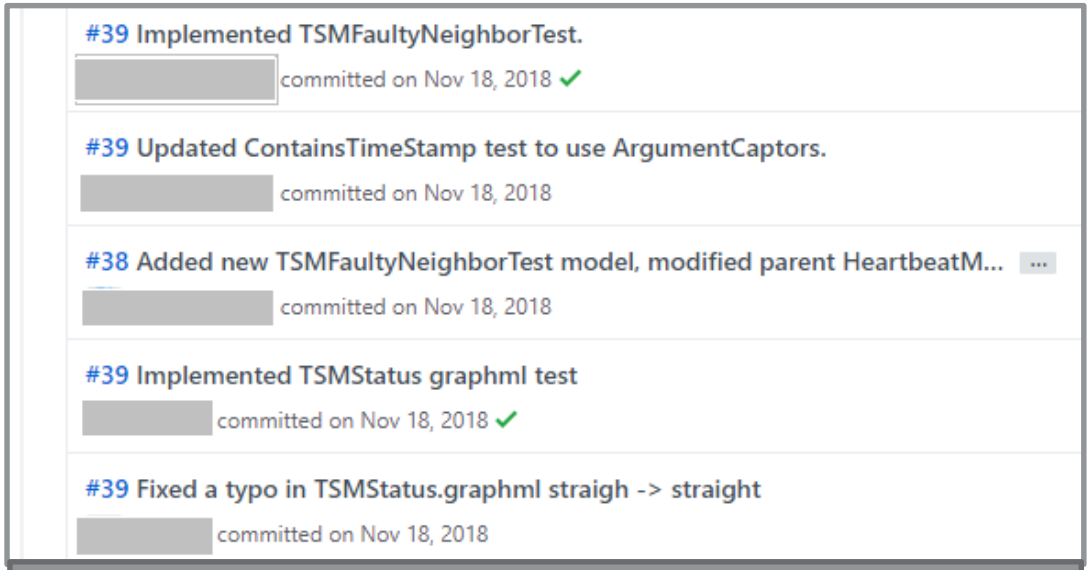
Local decision (Issue #20):

- Issue #20 detailed solution page
- Pull Request

Distributed decision (Issue #21):

- Distributed Decision Solution
- Pull Request

Wiki: organized



Clean code & commit history

Our opinion about MBT

The Model Based Testing approach looks interesting and pretty useful in test development. We really liked the idea of graph based testing for state machines, however, it has it's limits, for example working with timers is not so easy. Our experience with GraphWalker and yEd is mixed, we encountered a few limitations and inconsistency during the exercise in these tools, so the work wasn't so smooth.

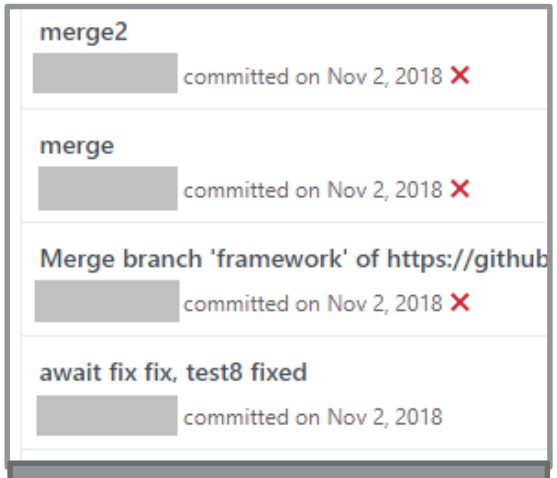
Documenting decisions and evaluation

Example home assignments – BAD

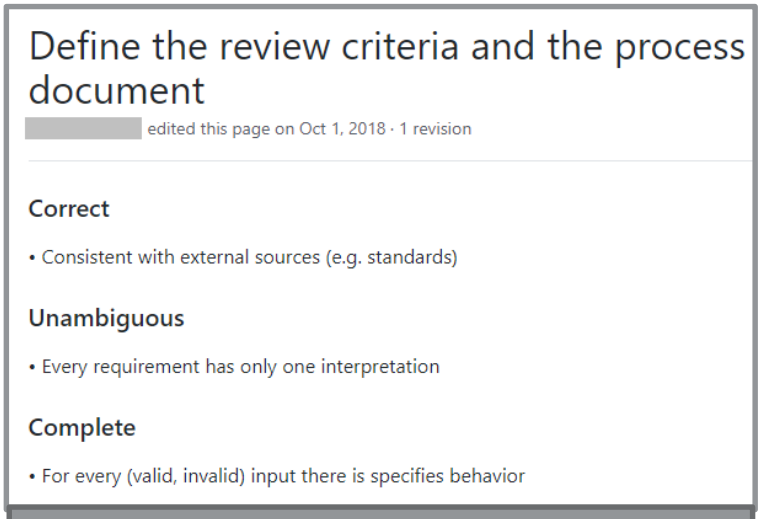
Finding the main problems (Good/Fair/Poor): Poor

- No commented SRS version is linked to the submission page. No file was uploaded to the master branch.

Artifacts to grade not linked/commited



Broken commits, no clean code



No explanations, no details