



Technische
Universität
Braunschweig



Projekt SPL Development, WS 17/18

Organization

Michael Nieke, Sven Schuster

- 5 Credits
- Scheine
 - Teamproject (Bachelor);
 - Softwaretechnik, vertiefendes Praktikum (Master, benotet);
 - Fahrzeuginformatik Praktikum (Master, unbenotet)
- Compulsory attendance
- Final presentation
- 11.09.2017 – 29.09.2017

- Each day: 9:00 – 17:00
- Room IZ 416
- Stand-up meetings at 10:00
- Make appointments with us!
- Maintain burndown chart with requirements/functionalities

- Pair programming
 - Switch teams each 3 days
 - Scrum with proper roles → we are project owner, one of you is scrum master
- GitHub project: <https://github.com/TUBS-ISF/ProjectSPLDevelopment>
 - Send Git username to: nieke@isf.cs.tu-bs.de
 - Use a new branch for each feature!
 - After finishing a feature make a merge from master branch and a pull request for the master!
- Dynamic requirement requests
- Keep configurations stable!

Documentation

- Documentation of experiences is crucial!
- Templates to document specific things (see next slides)
- Lessons learned? Established best practices / team rules? → Document!

Documentation

Problem Space Evolution

Feature Model Evolution:

- How did you evolve the FM?
- Why did you evolve the FM?

Same for:

- Mapping Evolution
- Configuration Evolution
- Context Evolution

Did you co-evolve FMs, mappings, configurations and context?

- How did you co-evolve them?
- Why did you co-evolve them?

Documentation

Evolution Inconsistencies

Did your FM evolution result in inconsistent configurations?

- When did you notice this?
- Why did it happen?
- How did you repair it?

Did your FM or mapping evolution result in different products (different deltas applied)?

- When did you notice this?
- Why did it happen?
- How did you repair it?

Did your FM or context evolution lead to inconsistent context-sensitivity (e.g., certain context values could not be set?)

Documentation

Evolution Inconsistencies

Did your FM or context evolution lead to inconsistent context-sensitivity (e.g., certain context values could not be set?)

- When did you notice this?
- Why did it happen?
- How did you repair it?

Did you break other code/interfaces by applying your delta?

- Did you break your own code or other developer's code?
- How did you repair it?
- Did you introduce conflicting realization artifacts?

Did you program against a different interface than you wanted to?

- Why? Did you use a wrong configuration premise?
- Were there redundancies and you chose the wrong one?

Whenever you perform a refactoring across delta borders...

- Why did you perform the refactoring?
- What exactly did you refactor? (e.g., method moved, deltas merged, deltas split, ...)