Next Weeks’ Schedule

Week 1 (Oct 17 – Oct 21)
■ Introduction lectures

Week 2 (Oct 24 – Oct 28)
■ Find teams, enroll!
■ Work on exercise
■ Lecture on Scrum
  □ Exercise after lunch!

Week 3 (Oct 31 – Nov 4)
■ POs: Customer meeting
■ Work on exercise

Week 4 (Nov 7 – Nov 11)
■ Kick-off presentation
■ Start of project
Your Project

Arian Treffer
arian.treffer@hpi.de

Software Engineering II
WS 2016/17

Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts group
SWT2 Schedule

LECTURES
- October 21
- October 28
- November 11
- December 2
- December 16
- February 10
- Guest lectures tba

PROJECT
- November 11: Kick-Off
- Nov 14 – Nov 18: Begin of Sprint 1
- Nov 28 – Dec 2: Begin of Sprint 2
- Dec 12 – Dec 16: Begin of Sprint 3
- December 16: Intermediate Presentation
- Dec 19 – Jan 1: Christmas Break
- Jan 16 – Jan 29: Begin of Sprint 4
- Jan 30 – Feb 3: Begin of Kanban Week
- February 10: Final Presentation
Schedule

- Exact dates should be **negotiated** with your tutor
  - Enter dates and rooms in public calendar
- Sprint reviews and plannings for next sprint can be merged
- PO should roughly know what the team has done **before the review**!
- **On demand**: User Research with Customer
Let’s get started

- **POs**
  - Meet with the customer
  - Extract Requirements + create user stories (Github Tickets)
  - Prepare Sprint planning

- **Teams**
  - Prepare working environment
  - Clone repository, try to get application working, understand architecture, ideally: play around
  - Find a regular timeslot for your meetings

- **Within the team + tutors**
  - Enter into Google Calendar until Nov 6
  - 1st Sprint Planning † CW 46 (09. – 13.11.)
Communication Infrastructure

- Email lists
  - Separate lists for each team (*lists.myhpi.de*)
  - Keep your teammates in the loop
  - Rules and filters help organizing your inbox

- [https://swt2-2016.slack.com](https://swt2-2016.slack.com)

- Wiki for lean and globally accessible documentation

- Ticket system for overview and feedback about current tasks and progress

- Telephone, Skype, IRC, ... and personal contact for direct communication

- ... be creative! (but let us know, we are interested in learning what might be useful in the future)
Google Calendar

- Advantages:
  - Available Everywhere
  - Easy Integration with Outlook & iCal
- Overview of team appointments
- Access granted by our tutors
The Swiss army knife of software development

- Integrating tools for most common activities in one place
- Wiki, Bug Tracking, Time Management, Project Analytics, Discussions, ...

Examples:
- MS Team Foundation Server
- Codebeamer
- Trac
- Redmine, Plan.io (SaaS based on Redmine)
- Github
Github Project

https://github.com/hpi-swt2/workshop-portal/
Scrum in GitHub – waffle.io

Version Control System

Repository to store software configuration items

Features:

- Versioning
- Dealing with variants: branches
- Access control
  - Authentication, authorization
  - Locking
  - Concurrent development
- Reporting
  - How many versions, variants, changes, persons
  - History of changes
How to check continuously that your software works?
Continuous Integration!

- Connected to version control
- Customizable run scripts
- Ideally covering all development branches
- Examples:
  - CruiseControl
  - Anthill
  - Jenkins/Hudson
  - Travis-CI
Travis CI

How to have a running version available at all times?
Deploy your app!

- Simple solution: local deployment
- Dedicated Servers, Infrastructure-as-a-Service, Platform-as-a-Service
  - more about that in a separate lecture
- We use Heroku (PaaS)
- Deployment is automatically triggered by successful Travis-CI build
http://heroku.com/
How to ensure that your software adheres to certain quality standards (complexity, test coverage, etc.)?

- Self-control
- Code Reviews (future lecture)
- Automatic checking
  - CodeClimate
  - SimpleCov
    - runs automatically during each test run
    - coverage/index.html in your application folder
CodeClimate

All Files (100.0% covered at 1.35 hits/line)

6 files in total. 41 relevant lines. 41 lines covered and 0 lines missed

<table>
<thead>
<tr>
<th>File</th>
<th>% covered</th>
<th>Lines</th>
<th>Relevant Lines</th>
<th>Lines covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>app/controllers/application_controller.rb</td>
<td>100.0 %</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>app/controllers/job_offers_controller.rb</td>
<td>100.0 %</td>
<td>77</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>app/helpers/application_helper.rb</td>
<td>100.0 %</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>app/helpers/job_offers_helper.rb</td>
<td>100.0 %</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>app/models/job_offer.rb</td>
<td>100.0 %</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>app/models/user.rb</td>
<td>100.0 %</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Your Project

1. Schedule
2. Calendar
3. Slack
4. Github
5. Travis CI
6. Heroku
7. CodeClimate
Next Weeks’ Schedule

Week 1 (Oct 17 – Oct 21)
- Introduction lectures

Week 2 (Oct 24 – Oct 28)
- Find teams, **enroll**!
- Work on exercise
- Lecture on Scrum
  - Exercise after lunch!

Week 3 (Oct 31 – Nov 4)
- POs: Customer meeting
- Work on exercise

Week 4 (Nov 7 – Nov 11)
- Kick-off presentation
- Start of project
Next Weeks’ Team Meetings

Week 5 (Nov 14 – Nov 18)
- PO presents team vision
- Estimation
- Begin 1st sprint
- Planning

Week 6 (Nov 21 – Nov 25)
- Weekly (daily) Scrum
- Estimation

Week 7 (Nov 28 – Dec 2)
- Finish 1st sprint
  - Review
  - Retrospective
- (Estimation)
- Begin 2nd sprint
  - Planning