Your Project

Christoph Matthies
christoph.matthies@hpi.de

Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts
Project

- Sprint 1: 9.11. (Kick-Off) - 23.11.
- **Weekly Stand-ups!**
- Sprint 1 review & retrospective + Sprint 2 planning: 23.11.

Meetings

- Exact dates for meetings should be *negotiated* with your tutor
- Sprint reviews and plannings for next sprint can be merged

- On demand: User Research with Customer
Let’s get started

POs
- Extract requirements + create user stories (GitHub tickets)
- Get an idea of the interaction workflows in the system (mockups?)
- Prepare Sprint plannings, inform yourselves on what the team is doing
- PO should roughly know what the team has done before the review

Developers
- Clone repository, get application working, understand architecture
- If you were lead architect, how would you construct this?
  - What are some challenges with the problem domain?
- Play around and try things out.
  - Where does the system have problems?
  - What makes no sense to you?
Let’s get started

SMs
- Part of your job is research and retrospection
  - What is working well in the team? What isn't?
  - What are good ways of how a team meeting can be structured?
- Every team is different. **Experiment!**

Meetings
- Regular meeting + work timeslot
- We have reserved **V1.15** and "Glaskasten" Fridays **11:00-17:00** for you

**Tutorium in this space after the lecture**
- Ask any (Ruby [on Rails]) question
- Work together
Project Infrastructure
VMware vSphere

VMware's cloud computing virtualization platform


- It allows managing VMs and resources in an online interface
- The URL as well as the user and password were mailed to you
- The application uses a library called 'rbvmomi' to access vSphere (see the Gemfile)
- The vSphere Center is only available within the HPI network!
VMware vSphere

Overall compliance status: Non-Compliant

Baseline: Upgrade to 6.5
Type: Host Upgrade
Compliance Status: Non-Compliant
Communication Infrastructure

- Email lists
  - If you want: separate lists for each team (lists.myhpi.de)
  - Keep your teammates in the loop, add teaching team
  - Rules and filters help organizing your inbox
- [https://swt2-2018.slack.com](https://swt2-2018.slack.com)
- Ticket system for overview and feedback about current tasks and progress
- Wiki for lean and globally **accessible documentation**
- Telephone and personal contact for direct communication
- ... be **creative**!
  (but let us know, we are interested in learning what might be useful in the future)
Time Management

Shared Calendar
- Available Everywhere
- Integration with Outlook & iCal
- Overview of team appointments
- SWT2 calendar access granted by teaching team
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:
- Microsoft Team Foundation Server
- Redmine, Plan.io (SaaS based on Redmine)
- Gitlab
- GitHub
Version Control System

Repository to store development artifacts

Features:
- Versioning
- Dealing with variants: branches
- Access control
  - Authentication, authorization
  - Locking
  - Concurrent development
- Reporting and communication
  - How many versions, variants, changes, persons
  - History of changes
Continuous Integration

How do you make sure your software always works?

Continuous Integration!
- Connected to version control
- Customizable run scripts
- Ideally covering all development branches
- Examples:
  - CruiseControl
  - Anthill
  - Jenkins/Hudson
  - Travis CI
<table>
<thead>
<tr>
<th>Project</th>
<th>Status</th>
<th>Description</th>
<th>Branch</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vm-dev</code></td>
<td>✅</td>
<td>Added storage and name fields to request and changed field order</td>
<td>Max Schneider</td>
<td>#135</td>
</tr>
<tr>
<td><code>login-dev-GH-34-35-36</code></td>
<td>✗</td>
<td>Display correct error messages when login fails f35 f36</td>
<td>Marius Lichtblau</td>
<td>#133</td>
</tr>
<tr>
<td><code>vm-dev</code></td>
<td>✅</td>
<td>Merge pull request #44 from hpi-swt2/Issue/7-VM-Auflistung</td>
<td>Max Schneider</td>
<td>#132</td>
</tr>
<tr>
<td><code>login-dev-GH-34-35-36</code></td>
<td>✗</td>
<td>Display correct error messages when login fails f35 f36</td>
<td>Marius Lichtblau</td>
<td>#130</td>
</tr>
<tr>
<td><code>Issue/7-VM-Auflistung</code></td>
<td>✅</td>
<td>Added rejection info, so I can give a reason why request was rejected</td>
<td>Max Schneider</td>
<td>#128</td>
</tr>
<tr>
<td><code>login-dev</code></td>
<td>✅</td>
<td>Add login page (45)</td>
<td>Marius Lichtblau</td>
<td>#127</td>
</tr>
<tr>
<td><code>login-dev-GH-32</code></td>
<td>✅</td>
<td>Fix style issues</td>
<td>Marius Lichtblau</td>
<td>#125</td>
</tr>
</tbody>
</table>
How can you always have a running version available? (why would you want to?)

Deploy your application!

- Simple solution: test deployment on local machine
- Deployment on separate machine:
  - Dedicated Servers
  - Infrastructure-as-a-Service
  - Platform-as-a-Service, e.g. Heroku

- Continuous Deployment:
  Deployment automatically triggered by successful CI build
  - Deployment config is part of the project
  - No extra effort
How can you ensure that the software adheres to certain quality standards (complexity, test coverage, etc.)?

Check for compliance

- Self-control, code reviews
- **Automatic** checks
  - Hosted tools: e.g. CodeClimate, Codefactor, Codebeat
  - Local code coverage: SimpleCov (http://www.simplecov.org/)
    - Can run automatically during each test run
    - coverage/index.html in your application folder
  - Local code smells: RuboCop (https://www.rubocop.org)
Hosted Code Quality Checks

Breakdown

161 FILES

Codebase summary

MAINTAINABILITY

4 days

Repository stats

CODE SMELLS

29

DUPICATION

12

OTHER ISSUES

0
SimpleCov Code Coverage

**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. VMware vSphere
2. Communication infrastructure
3. Continuous Integration
4. Continuous Deployment
5. Code quality

6. Any other tools you want to add
   But, your team is not the only one using it, communicate.
Image Credits

- "Scrum process" by Lakeworks - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0-2.5-2.0-1.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Scrum_process.svg#mediaviewer/File:Scrum_process.svg