Next Weeks' Schedule

Week 1 (Oct 12 – Oct 16)
- Introduction lectures

Week 2 (Oct 19 – Oct 23)
- Find teams, **enroll**!
- Code School exercise
- Lecture on Scrum
  - ☐ Exercise after lunch!

Week 3 (Oct 26 – Oct 30)
- POs: Customer meeting
- Ruby on Rails exercise
- Lecture on Git and testing

Week 4 (Nov 2 – Nov 6)
- Kick-off presentation
- Start of project

Your Project

Arian Treffer
arian.treffer@hpi.de

Software Engineering II
WS 2015/16

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

Schedule

- **Exercises**
- **Kick-Off presentation**
- **Sprint 1 (2 weeks)**
- **Sprint 2 (2 weeks)**
- **Intermediate presentation**

- **Sprint 3 (3 weeks)**
- **Sprint 4 (2 weeks)**
- **Kanban Week**
- **Final presentation**
- **Exam (date tba)**

Oct	42	12	13	14	15	16	Lecture
	43	19	20	21	22	23	Lecture / Scrum Exercise
	44	26	27	28	29	30	Lecture
	45	2	3	4	5	6	Kick-Off / Lecture
Nov	46	9	10	11	12	13	
	47	16	17	18	19	20	
	48	23	24	25	26	27	
Dec	49	30	1	2	3	4	Lecture
	50	7	8	9	10	11	
	51	14	15	16	17	18	Interm. Presentation / Lecture

2 weeks break

Jan	1	4	5	6	7	8	
	2	11	12	13	14	15	Guest Lectures …
	3	18	19	20	21	22	
	4	25	26	27	28	29	
Feb	5	1	2	3	4	5	Final Presentation / Summary

HPI

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 3
  - ☐ 1st Sprint Planning → CW 46 (09. – 13.11.)

Project Infrastructure

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://swt2-2015.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)

Time Management

Google Calendar

- Advantages:
  - Available Everywhere
  - Easy Integration with Outlook & iCal
- Overview of team appointments
- Access granted by our tutors

Application Lifecycle Management

The Swiss army knive 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/wimi-portal**

Scrum in GitHub – waffle.io

- https://waffle.io/hpi-swt2/wimi-portal

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
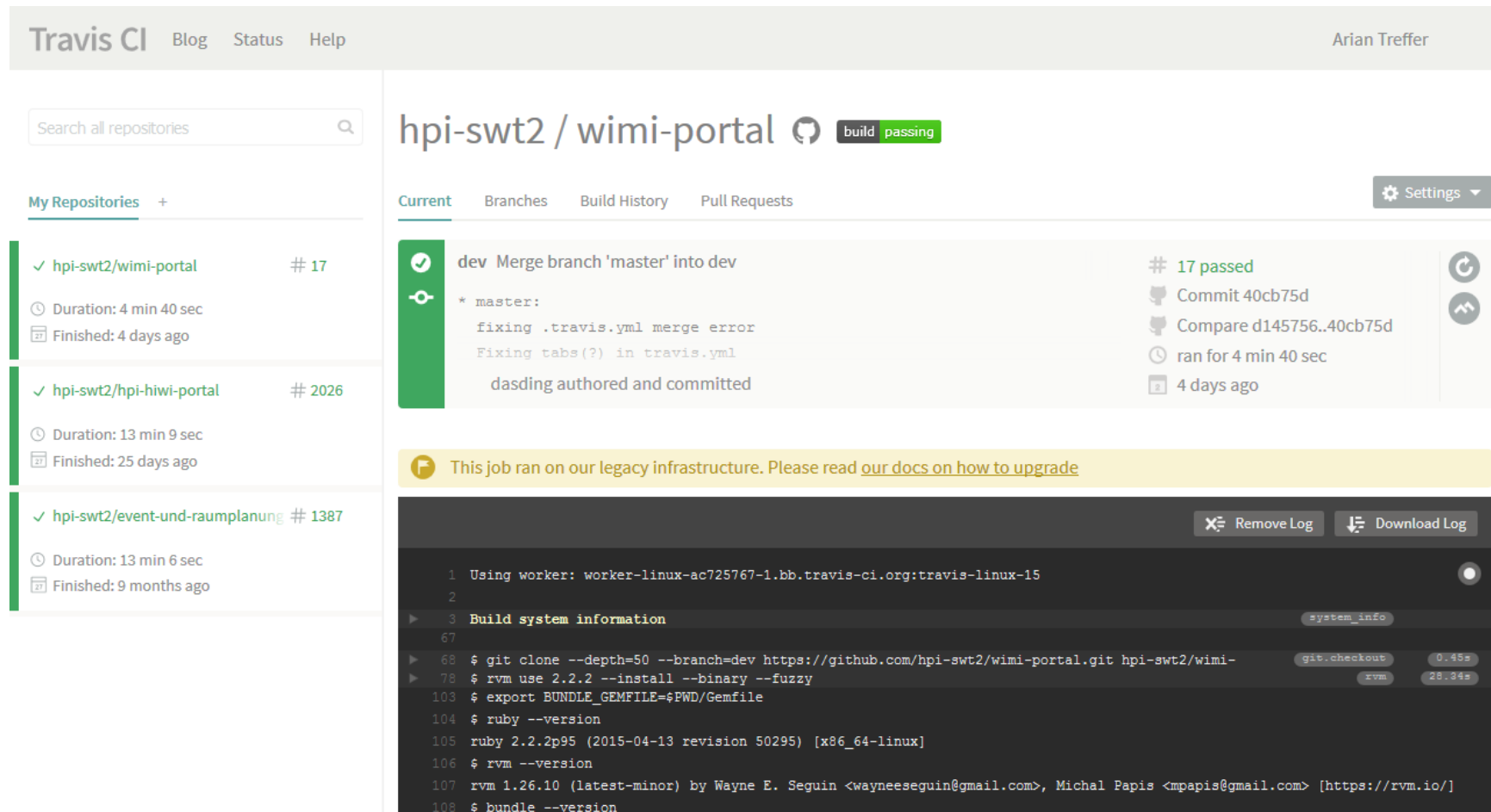 - History of changes

Continuous Integration

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

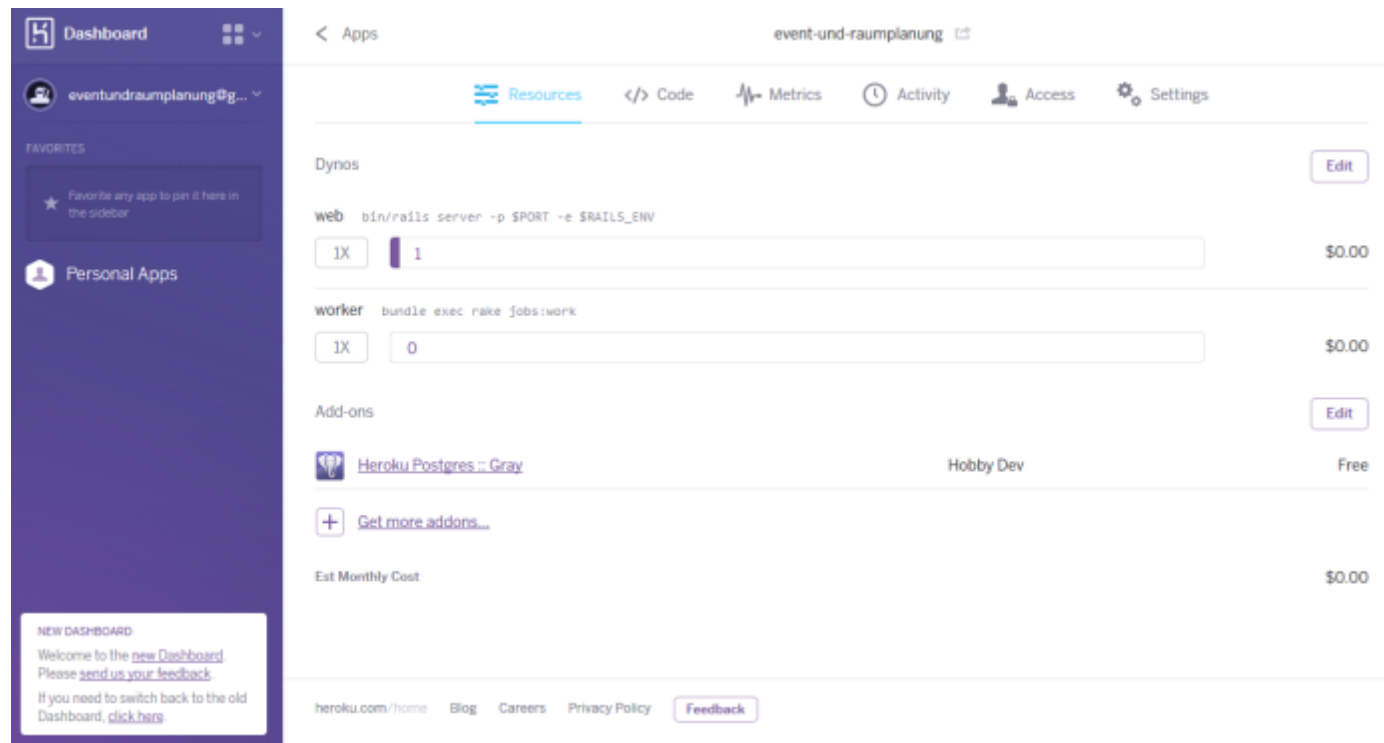■ https://travis-ci.org/hpi-swt2/wimi-portal

Application Deployment

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

Heroku

- http://heroku.com/
- http://wimi-portal.herokuapp.com/

Code Quality

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
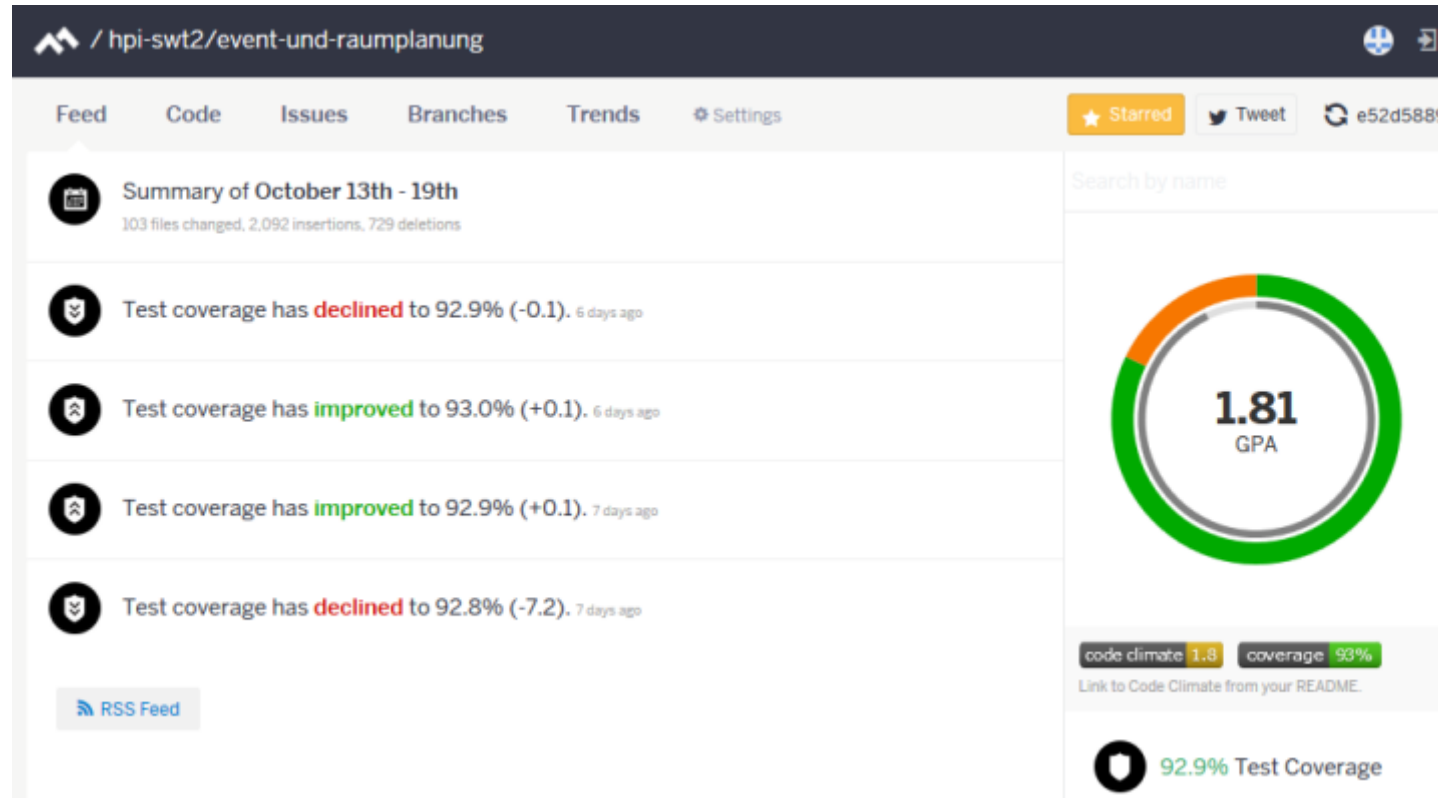
CodeClimate

- https://codeclimate.com/github/hpi-swt2/wimi-portal

SimpleCov

File	% covered	Lines	Relevant Lines	Lines cove
app/controllers/application_controller.rb	100.0 %	5	2	2
app/controllers/job_offers_controller.rb	100.0 %	77	34	34
app/helpers/application_helper.rb	100.0 %	2	1	1
app/helpers/job_offers_helper.rb	100.0 %	2	1	1
app/models/job_offer.rb	100.0 %	2	1	1
app/models/user.rb	100.0 %	7	2	2

Your Project

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

Image Credits

- "ST vs Gloucester - Match - 23" by PierreSelim - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:ST_vs_Gloucester_-_Match_-_23.JPG#mediaviewer/File:ST_vs_Gloucester_-_Match_-_23.JPG
- "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
- „Wien - Seestadt, SW-Areal 2013 (2)" von Bwag - Eigenes Werk. Lizenziert unter Creative Commons Attribution-Share Alike 3.0-at über Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Wien_-_Seestadt,_SW-Areal_2013_(2).JPG#mediaviewer/File:Wien_-_Seestadt,_SW-Areal_2013_(2).JPG
- "Utility pole in Curitiba" by Leonardo.stabile - Own work. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Utility_pole_in_Curitiba.JPG#mediaviewer/File:Utility_pole_in_Curitiba.JPG