

# Next Weeks' Schedule



## Week 1 (Oct 13 – Oct 17)

- Introduction lectures

## Week 2 (Oct 20 – Oct 24)

- Find teams, **enroll!**
- Code School exercise
- Lecture on Scrum
  - Important exercise after Lunch!
- Project Infrastructure

## Week 3 (Oct 27 – Oct 31)

- POs: Customer meeting
- Code School exercise
- No lecture

## Week 4 (Nov 3 – Nov 7)

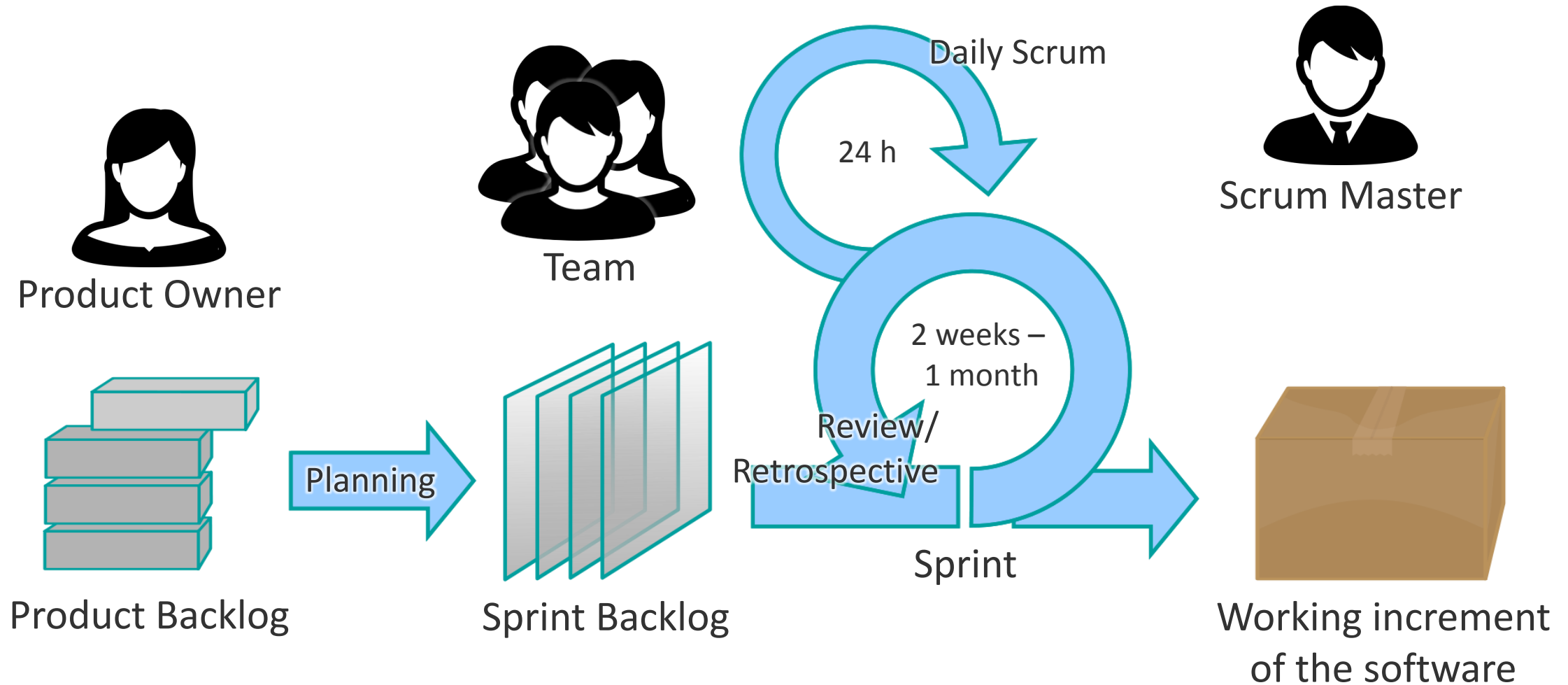
- PO Vision (~30min)
  - Mo 15:15
  - ... or propose other date
- Start of project



Scrum

Software Engineering II  
WS 2014/15

# Scrum



# Scrum



## Team

- Product Owner
- Scrum Master
- Developers

## Meetings

- Planning
- Daily Scrum
- Review
- Retrospective

## Artifacts

- Product Backlog
- Sprint Backlog
- User Stories
- Software Increment

# Effort, Schedule, and Cost Estimation



- Depends on software engineering process
- Highly **uncertain**, must be negotiated and revised with stakeholders
- Waterfall effort estimation
  - Methods: calibrated estimation model based on historical size (Function Points, LOC, ...); expert judgment; ...
  - Output: X man-months
- Agile effort estimation
  - **Iterative** methods, **shorter** planning horizon
  - Output: functionality to be implemented in the **next iteration**

# Requirements



In Scrum, requirements are usually defined as user stories:

“As <role>, I want <feature> to <reason>”

Stories need to fulfill **INVEST** properties:

- I – Independent
- N – Negotiable
- V – Valuable
- E – Estimatable
- S – Small
- T – Testable

<http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

# Tasks



For better planning, stories are broken down into tasks

Tasks should be **SMART**:

- S – Specific
- M – Measurable
- A – Achievable
- R – Relevant
- T – Time-boxed

<http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

# Effort Estimation in Scrum with “Planning Poker”



## Participants

- **Everyone** operationally involved in creating the software product
- Product owner (and Scrum Master) are not playing

## Preconditions

- Product backlog is complete and **prioritized**
- Backlog items are known by the team
- The effort for a small backlog item was determined as a **reference**
- Every participant has a set with sizing cards



# Planning Poker 1/2



- Product owner explains a backlog item
- Product owner **answers questions** of team members
- Every participant evaluates the complexity of the backlog item and chooses a card (**hidden**)
- All cards are shown simultaneously
- Participants with highest and lowest number **explain choices**
- The arguments are **discussed** in the group

# Planning Poker 2/2



- A new vote is conducted
- **Team agrees** on item size
  - Most occurring or average value is acceptable
  - If not, another round is played
- The moderator notes size of backlog item in the product backlog
- The game ends if all backlog items are sized or **time is over**

# After Planning Poker



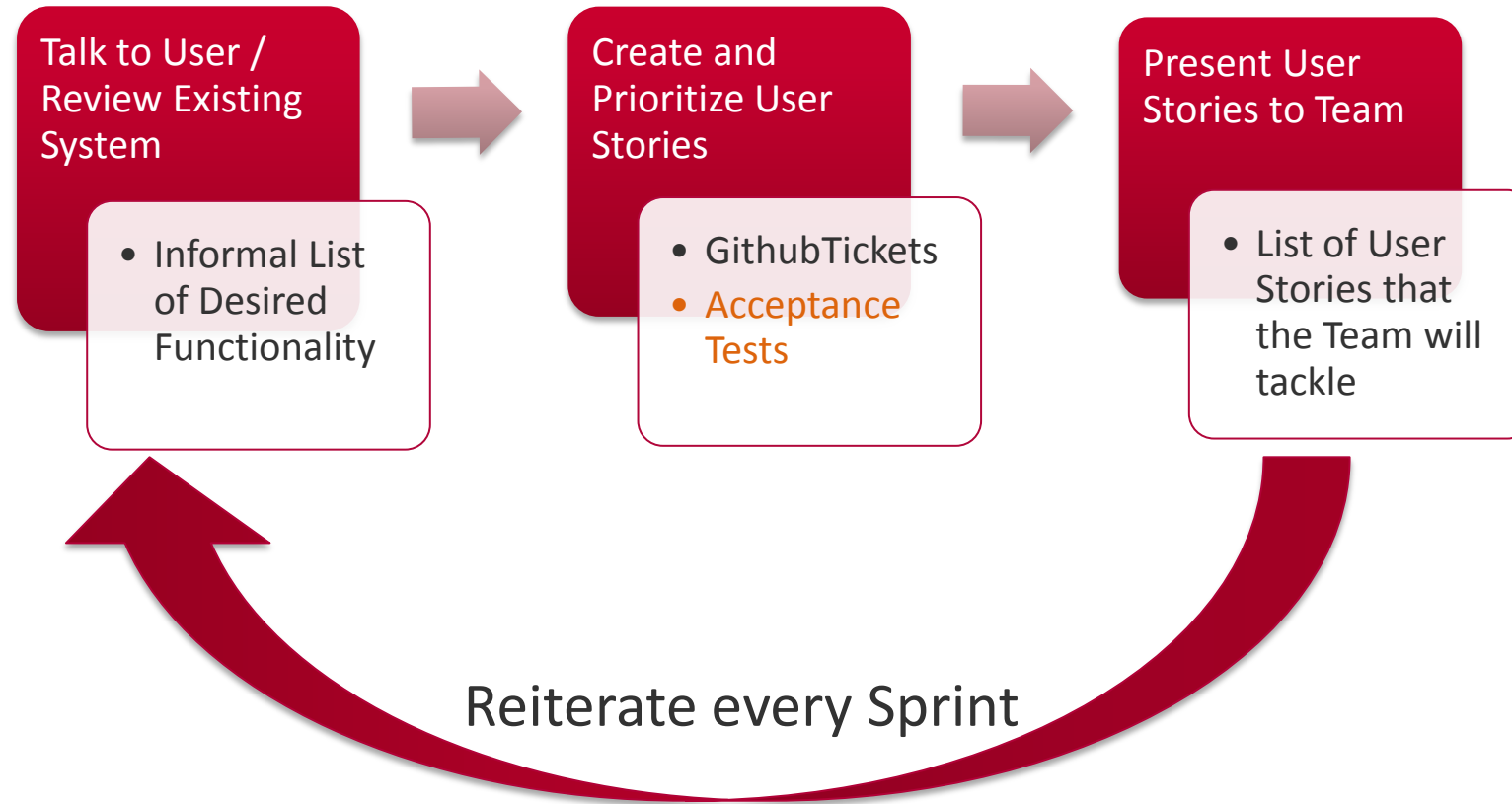
## Begin the sprint

- Select stories until sprint is full
- Break down stories into tasks and fill your **Scrum Board**
- Assign stories to developer(s)
- **Implement** the stories task by task

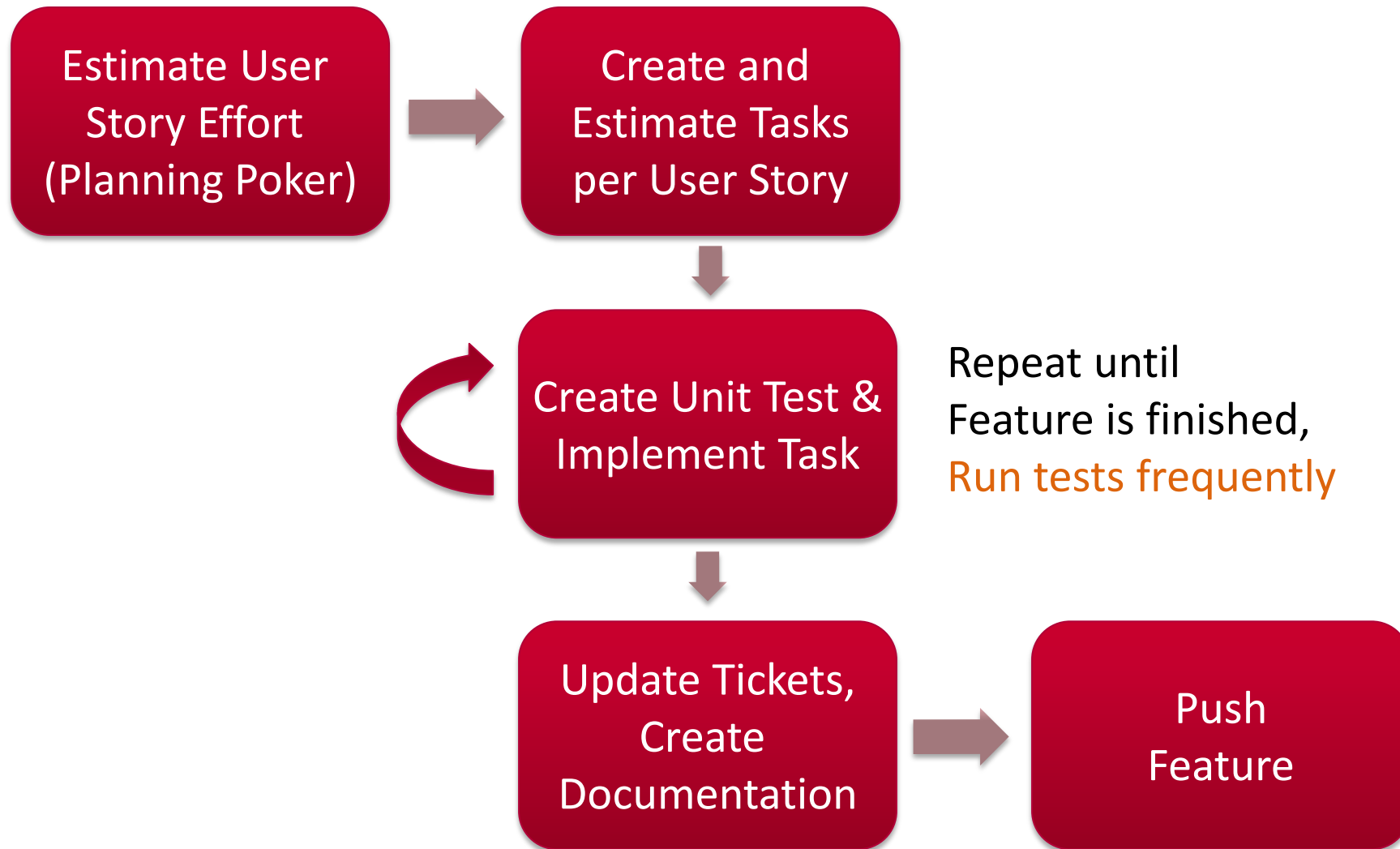
## Done and sprint is not over, yet?

- **Help** your teammates
- Refactor, **write tests**, document
- Ask the Product Owner for more work

# Projekt Workflow: Product Owner



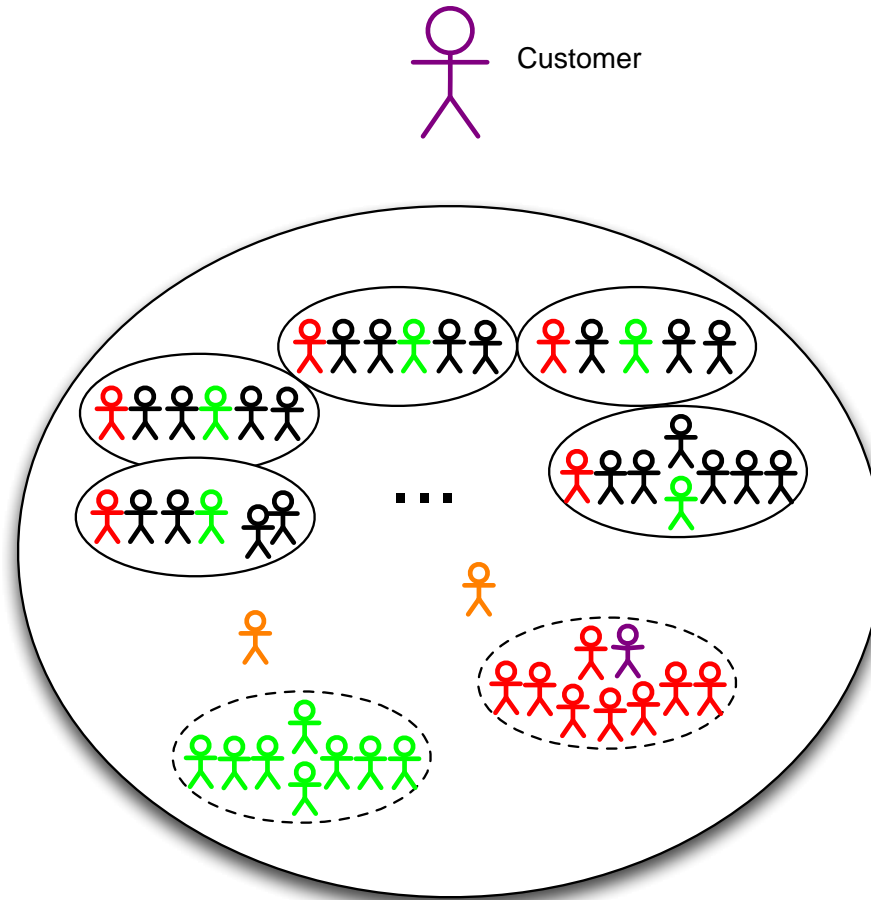
# Project Workflow: Developers





# Scaling Scrum

# Recap: High-level Overview of SWT2



# Implications of the Setup



What's needed in such an environment?

- Development **process**
- **Communication** on multiple levels
- Infrastructure for **collaboration**



# Scaling Scrum: Project Start



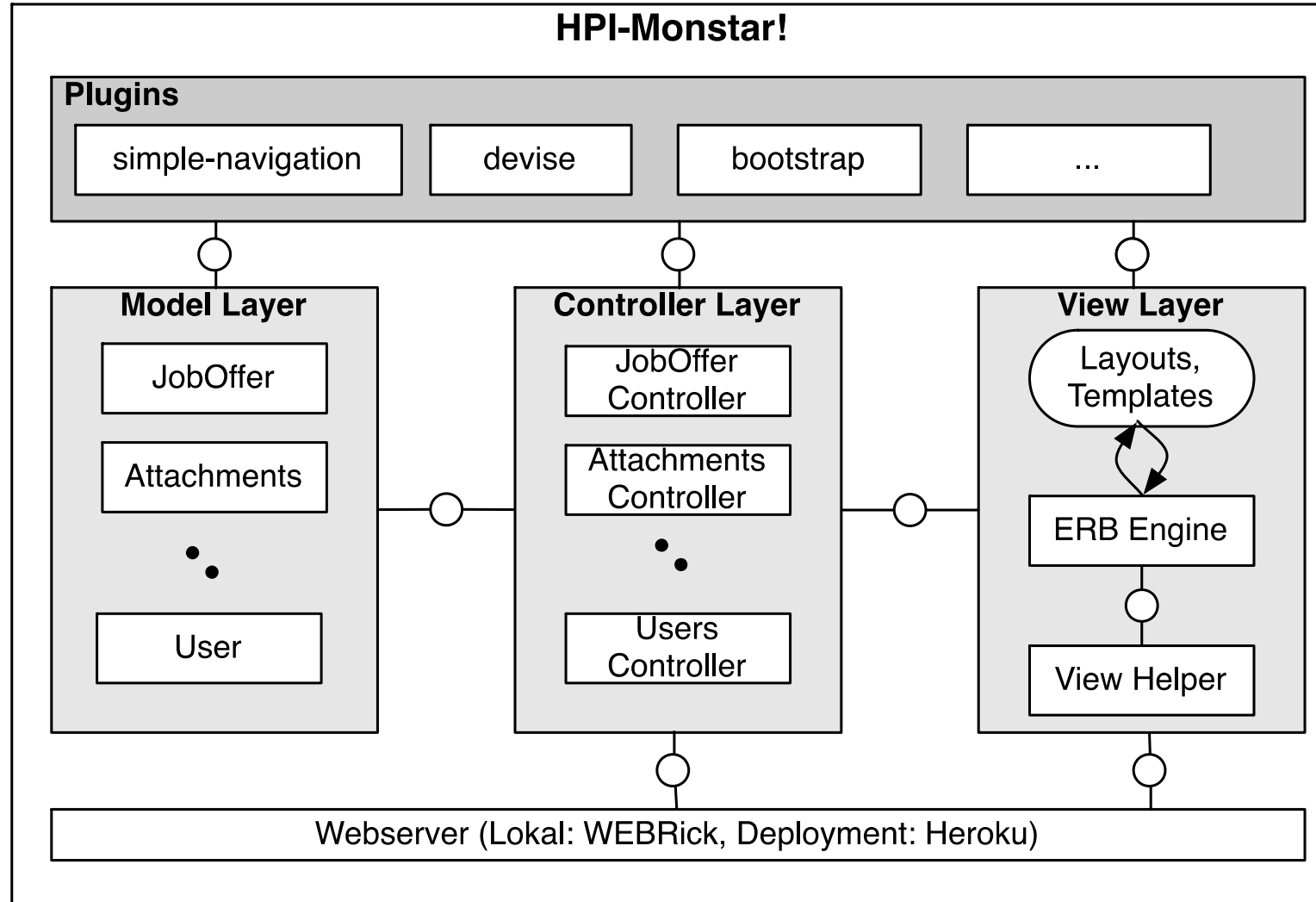
Start small and grow organically

- Single Scrum team for preparation
- Work out foundation for the first sprints
- Scale when it becomes necessary

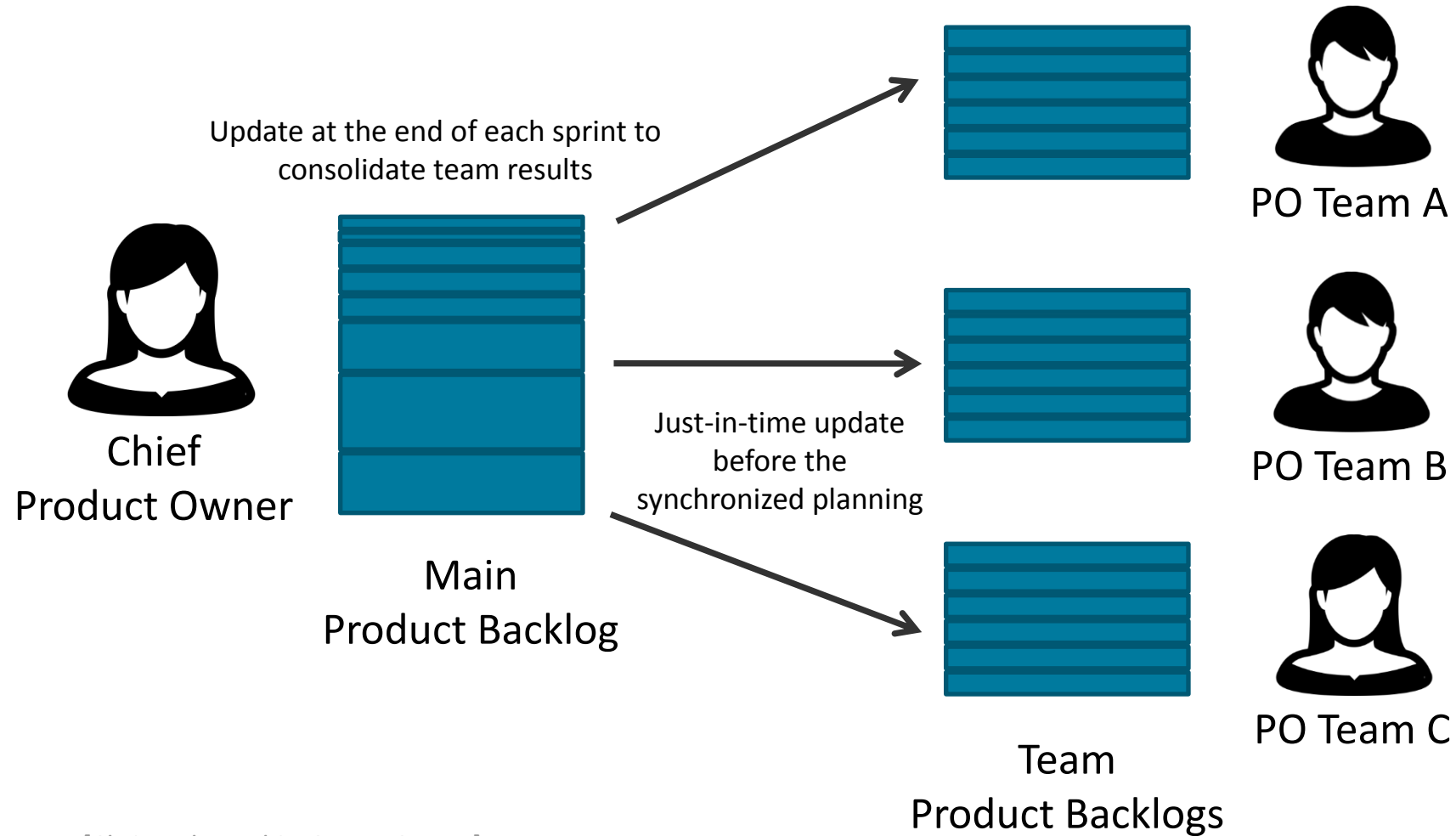
We are now at the first scaling point!

- Rudimentary architecture is present
- Infrastructure is prepared and **ready to go**

# Architecture Overview



# Product Owner / Backlog Hierarchy



[Christoph Mathis, Scrum Center]

# Scaling Scrum: Sprint Planning

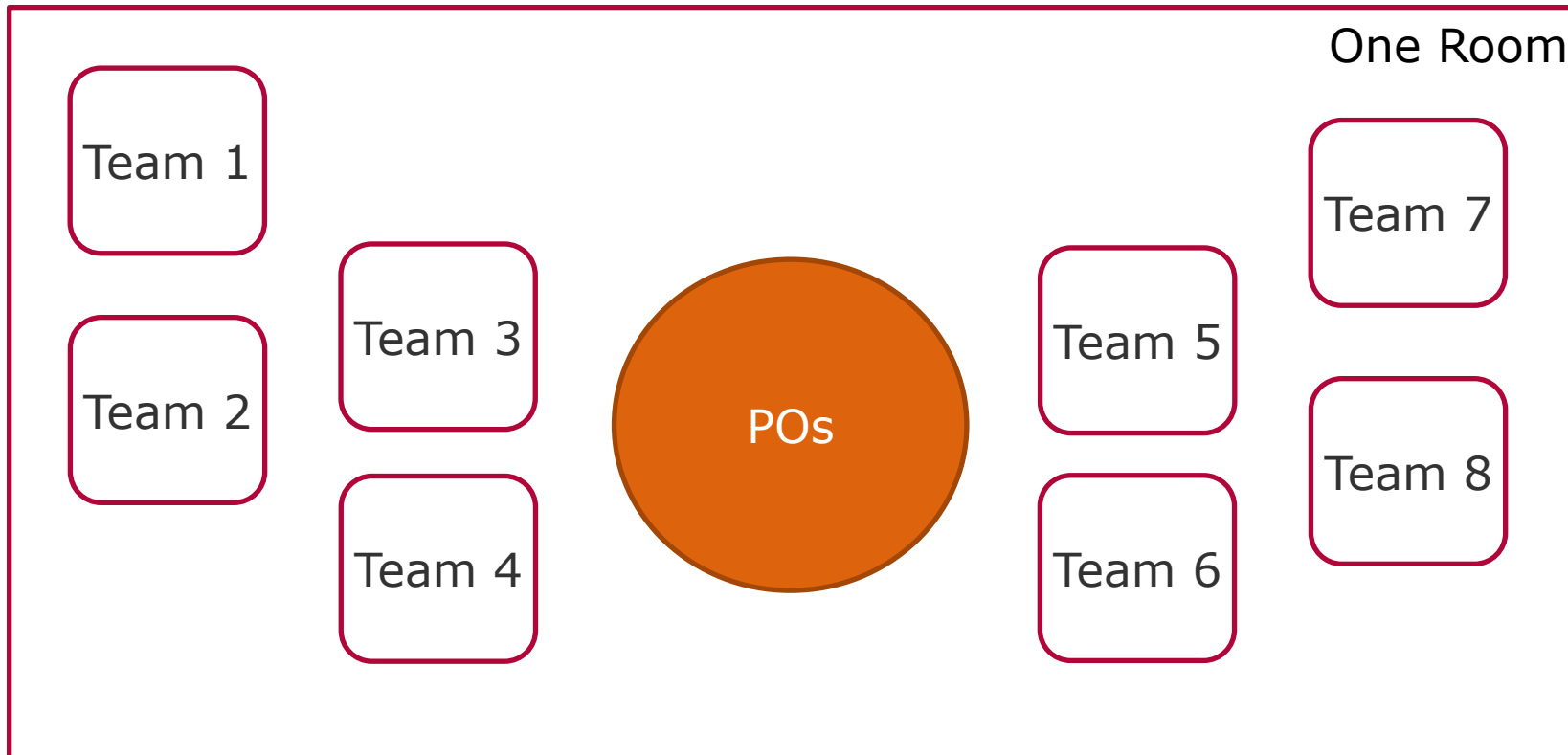


- Preparation
  - Individual review and retrospection meetings
  - Meeting of all teams with 1-2 members each:
    - Review of the last sprint
    - Input dependencies (What is needed)
    - Output dependencies (What needs to be delivered)
- Execution
  - Individual plannings (strict timeboxing)
  - Discussion of identified additional input or output dependencies
  - Final sprint planning
- Problem: Time consuming & high degree of coordination needed!

# Scaling Scrum: Sprint Planning



Another Option: Co-located planning





# Project Timeline

Software Engineering II  
WS 2014/15

# Project Timeline



Sprint 1  
CW 45 – 47

Sprint 2  
CW 48 – 50

Sprint 3  
CW 51 – 03

Sprint 4  
CW 04 – 06

- Exact dates should be **negotiated** with your tutor
- Sprint reviews and plannings for next sprint can be merged
- PO should roughly know what the team has done **before the review!**

# Important Dates



- 03.11. **Kickoff Presentation**
- CW 45 (03. – 07.11.): Planning Sprint #1
- CW 47 (17. – 21.11.): Review Sprint #1 / Planning Sprint #2
- CW 50 (08. – 12.12.): Review Sprint #2 / Planning Sprint #3
- 12.12. **Intermediate Presentation**
- CW 03 (12. – 16.01.): Review Sprint #3 / Planning Sprint #4
- CW 06 (02. – 06.02.): Review Sprint #4
- 06.02. **Final Presentation**, Course Retrospective, Exam Q&A
  
- **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 2
  - 1st Sprint Planning -> CW 45 (03. – 07.11.)



# Project Infrastructure

Software Engineering II  
WS 2014/15

# 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://swt2-2014.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 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-sw22/event-und-raumplanung>

The screenshot shows the GitHub repository page for 'hpi-sw22 / event-und-raumplanung'. At the top, there is a search bar and navigation links for 'Explore', 'Gist', 'Blog', and 'Help'. The repository name is displayed with icons for 'Unwatch' (4), 'Unstar' (3), and 'Fork' (1). Below the repository name, a description reads: 'Ein Tool das die interne Planung von Events verbessern soll und dabei besonderen Fokus auf die Zuteilung von Räumen und Ausstattung legt. — Edit'. A summary bar shows '77 commits', '1 branch', '0 releases', and '3 contributors'. The main content area shows the file tree for the 'event-und-raumplanung' branch on 'master'. A commit by 'chrisma' is highlighted with the message 'Use lazy loading of translations.' The file tree includes folders like 'app', 'bin', 'config', 'db', 'lib', 'log', 'public', 'test', and 'vendor/assets', each with a commit message and timestamp. On the right sidebar, there are links for 'Code', 'Issues' (1), 'Pull Requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. At the bottom of the sidebar, there are options to 'Clone in Desktop' and 'Download ZIP'.

File/Folder	Commit Message	Time Ago
app	Use lazy loading of translations.	a day ago
bin	Initial commit	12 days ago
config	Use lazy loading of translations.	a day ago
db	Remove redundant User views, devise already creates sign_up and edit.	8 days ago
lib	Initial commit	12 days ago
log	Initial commit	12 days ago
public	Initial commit	12 days ago
test	Removed user fixture (left over from previous scaffold) that caused t...	7 days ago
vendor/assets	Initial commit	12 days ago
nitonore	Ignore anvil files created by travis deploy	6 days ago

# Scrum in GitHub - waffle.io



- <https://waffle.io/hpi-swt2/event-und-raumplanung>

The screenshot shows a Kanban board with four columns: Backlog (0 items), Ready (0 items), In Progress (1 item), and Done (3 items). The 'In Progress' column contains one card titled 'Applikationsrumpf anlegen' with a 'Preface' label and 8 comments. The 'Done' column contains three cards: 'Unnötig' (with 'pull-request-5' and 'review.ninja' labels and 1 comment), 'Infrastruktur zusammeklicken' (with a 'Preface' label and 1 comment), and 'Example pull request for review.ninja' (with a 'Preface' label and 4 comments). The interface includes a top navigation bar with the project name, an 'Add Issue' button, and a 'Filter Board' search box. A left sidebar contains navigation icons for settings, charts, home, and help.

# Version Control System



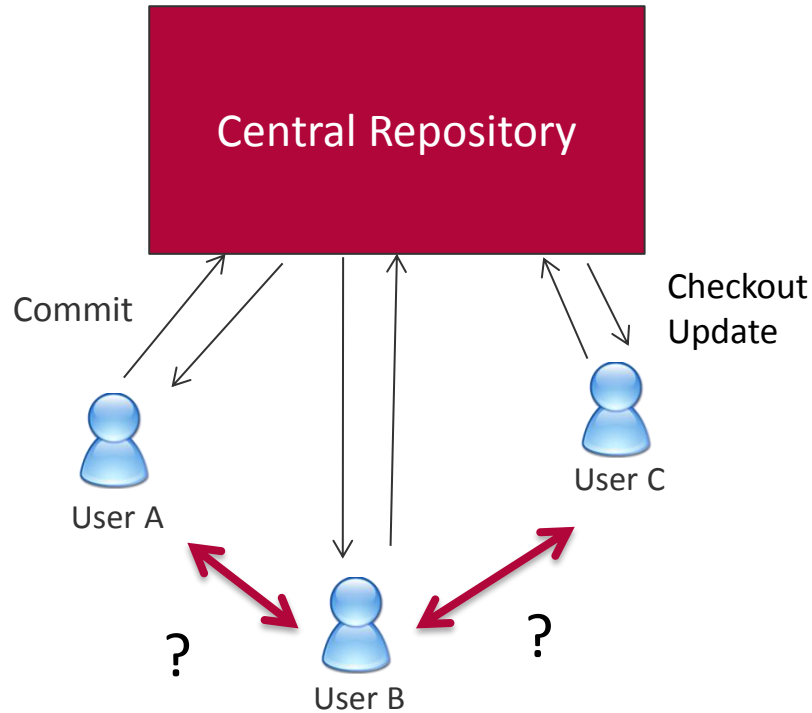
Repository to store software configuration items

Features:

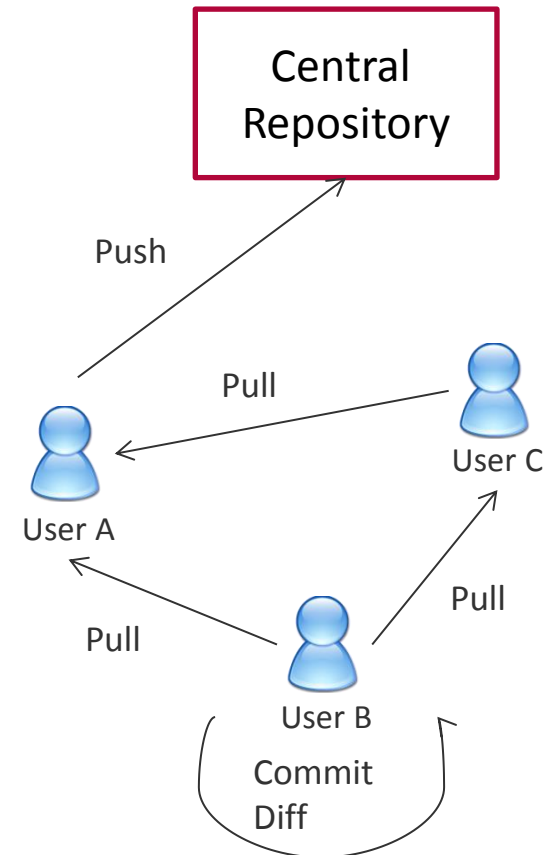
- Versioning
- Dealing with variants: **branches**
- Access control
  - Authentication, authorization
  - Locking
  - **Concurrent** development
- Reporting
  - How many versions, variants, changes, persons
  - History of changes



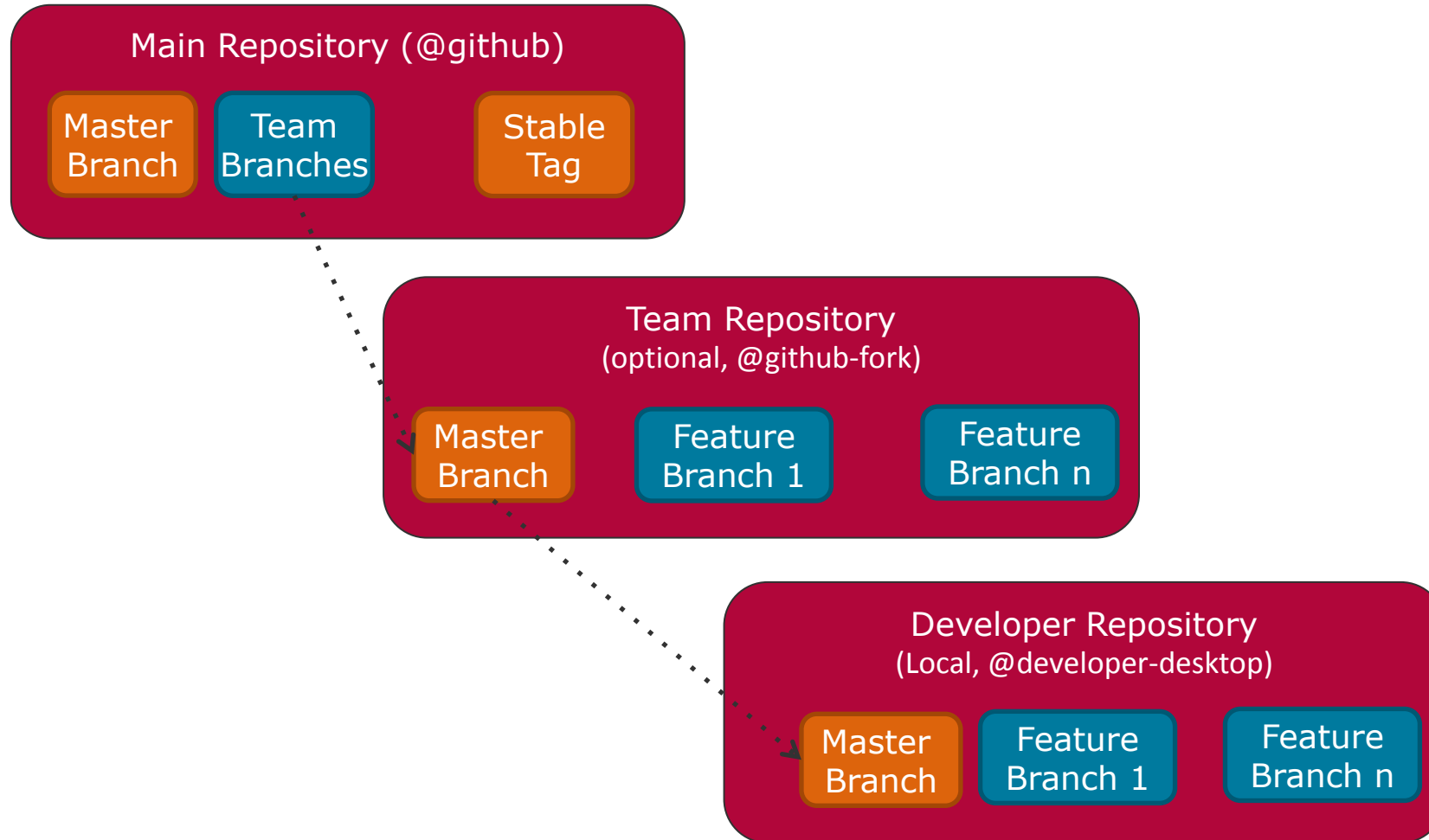
# Centralized vs Distributed VCS



VS.



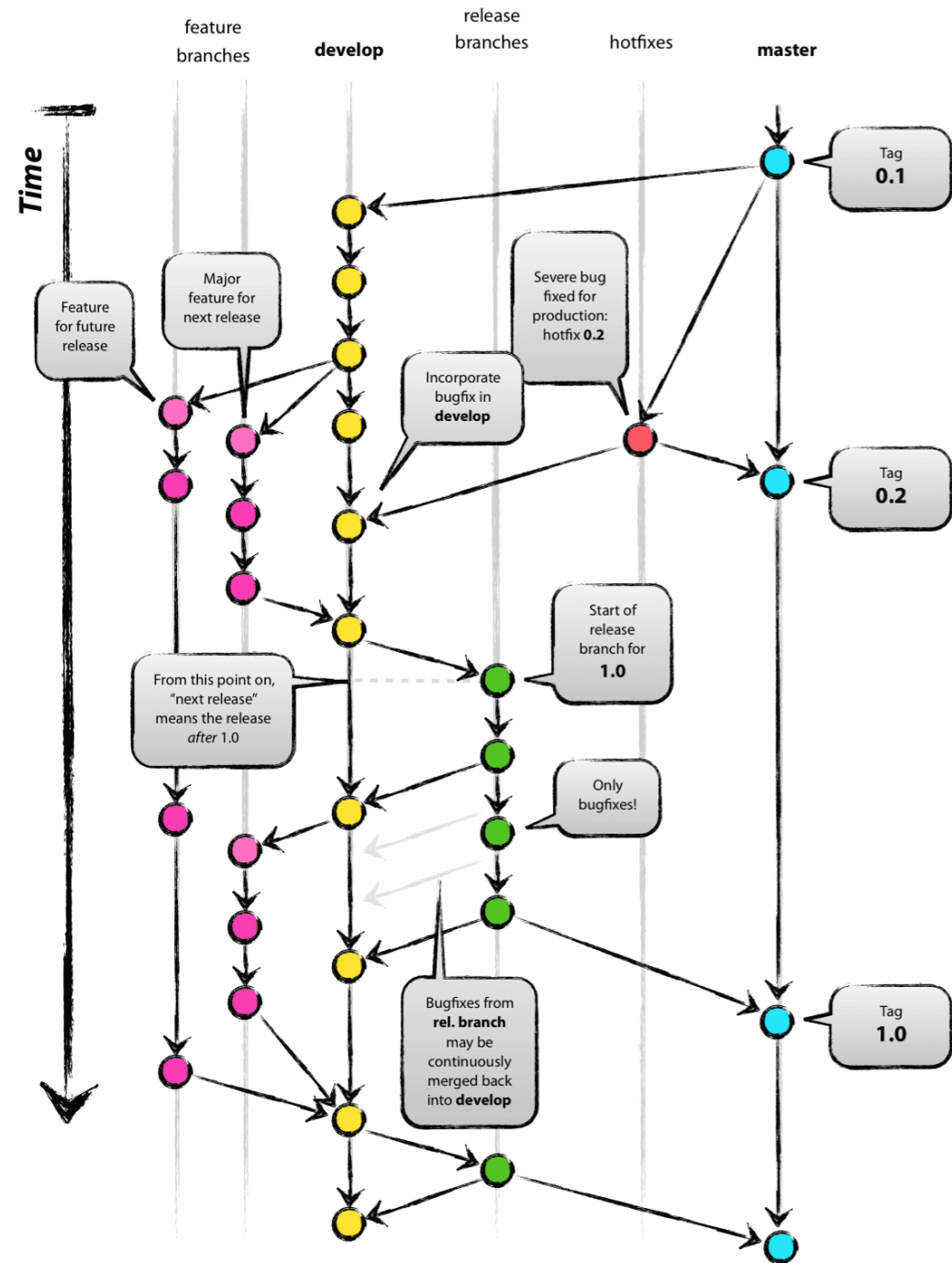
# Project Repository Setup



# Branching



- <http://nvie.com/posts/a-successful-git-branching-model/>



# Continuous Integration

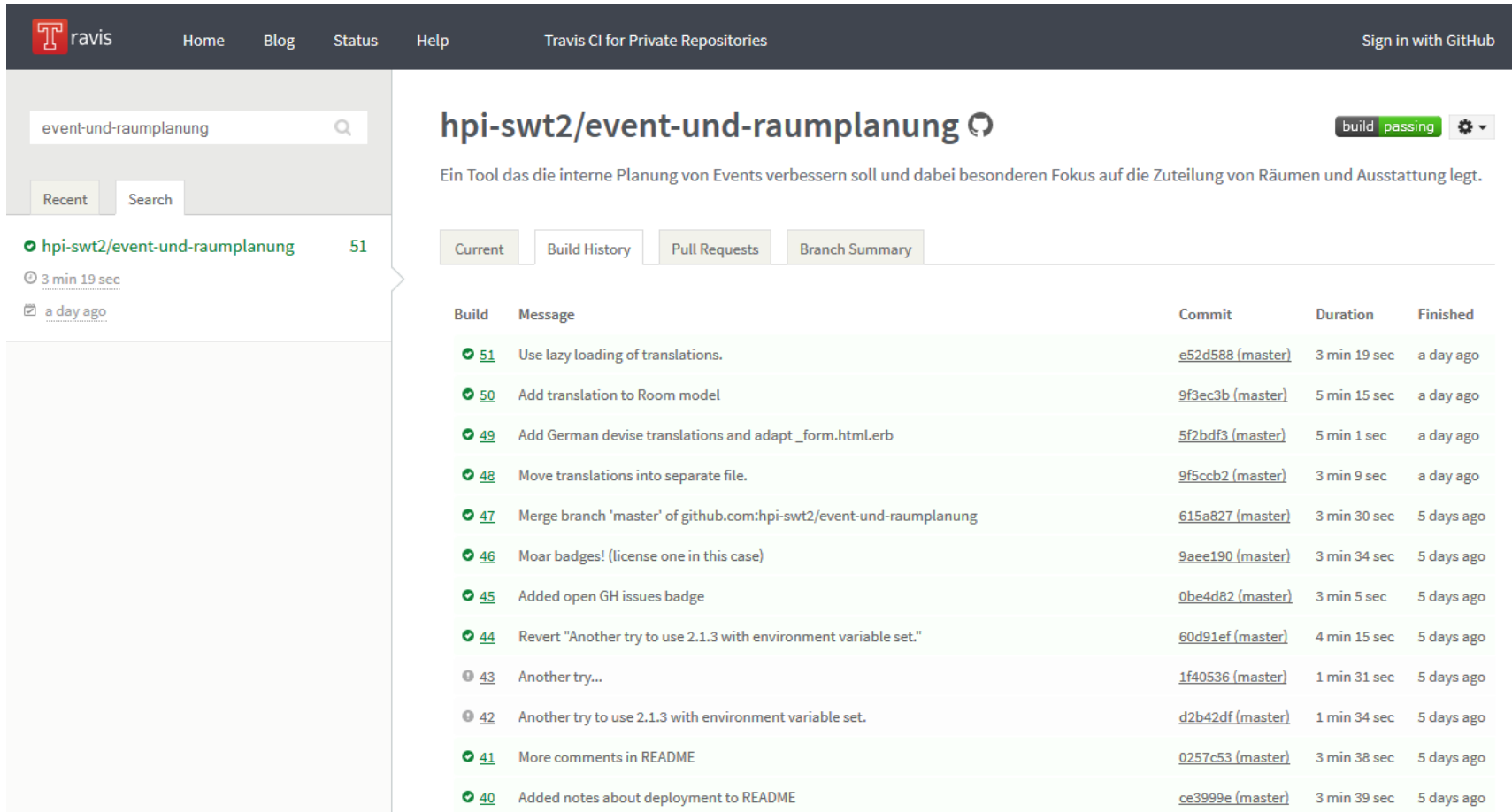


How to check **continuously** that your software works?

Continuous Integration!

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

## ■ <https://travis-ci.org/hpi-swt2/event-und-raumplanung>



The screenshot displays the Travis CI web interface for the repository `hpi-swt2/event-und-raumplanung`. The top navigation bar includes links for Home, Blog, Status, Help, and Travis CI for Private Repositories, along with a 'Sign in with GitHub' button. The main content area shows the repository name and a 'build passing' status. Below this, there are tabs for 'Current', 'Build History', 'Pull Requests', and 'Branch Summary'. The 'Build History' tab is active, showing a table of recent builds.

Build	Message	Commit	Duration	Finished
<a href="#">51</a>	Use lazy loading of translations.	<a href="#">e52d588 (master)</a>	3 min 19 sec	a day ago
<a href="#">50</a>	Add translation to Room model	<a href="#">9f3ec3b (master)</a>	5 min 15 sec	a day ago
<a href="#">49</a>	Add German devise translations and adapt _form.html.erb	<a href="#">5f2bdf3 (master)</a>	5 min 1 sec	a day ago
<a href="#">48</a>	Move translations into separate file.	<a href="#">9f5ccb2 (master)</a>	3 min 9 sec	a day ago
<a href="#">47</a>	Merge branch 'master' of github.com:hpi-swt2/event-und-raumplanung	<a href="#">615a827 (master)</a>	3 min 30 sec	5 days ago
<a href="#">46</a>	Moar badges! (license one in this case)	<a href="#">9aee190 (master)</a>	3 min 34 sec	5 days ago
<a href="#">45</a>	Added open GH issues badge	<a href="#">0be4d82 (master)</a>	3 min 5 sec	5 days ago
<a href="#">44</a>	Revert "Another try to use 2.1.3 with environment variable set."	<a href="#">60d91ef (master)</a>	4 min 15 sec	5 days ago
<a href="#">43</a>	Another try...	<a href="#">1f40536 (master)</a>	1 min 31 sec	5 days ago
<a href="#">42</a>	Another try to use 2.1.3 with environment variable set.	<a href="#">d2b42df (master)</a>	1 min 34 sec	5 days ago
<a href="#">41</a>	More comments in README	<a href="#">0257c53 (master)</a>	3 min 38 sec	5 days ago
<a href="#">40</a>	Added notes about deployment to README	<a href="#">ce3999e (master)</a>	3 min 39 sec	5 days ago

# Application Deployment

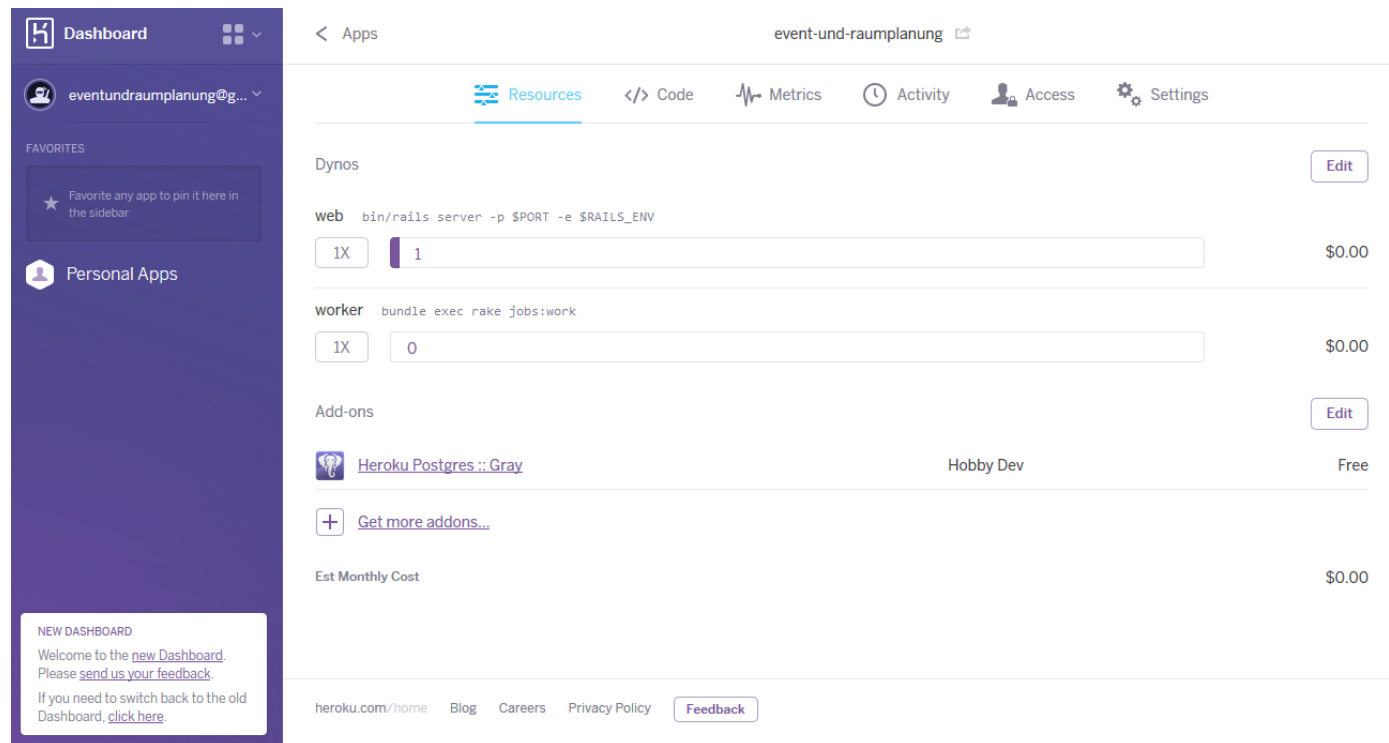


How to have a **running version** available at all times?

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

- <http://heroku.com/>
- <http://event-und-raumplanung.herokuapp.com/>



The screenshot shows the Heroku dashboard for the application 'event-und-raumplanung'. The interface includes a sidebar with 'Dashboard' and 'Personal Apps' sections. The main content area displays the application's configuration, including Dynos (1X web, 1 worker), Add-ons (Heroku Postgres :: Gray, Hobby Dev), and an estimated monthly cost of \$0.00. A 'NEW DASHBOARD' notification is visible in the bottom left corner of the dashboard area.

Dashboard

eventundraumplanung@g...

FAVORITES

Favorite any app to pin it here in the sidebar

Personal Apps

NEW DASHBOARD

Welcome to the new Dashboard. Please [send us your feedback](#). If you need to switch back to the old Dashboard, [click here](#).

Apps event-und-raumplanung

Resources Code Metrics Activity Access Settings

Dynos Edit

web bin/rails server -p \$PORT -e \$RAILS\_ENV

1X 1 \$0.00

worker bundle exec rake jobs:work

1X 0 \$0.00

Add-ons Edit

Heroku Postgres :: Gray Hobby Dev Free

+ Get more addons...

Est Monthly Cost \$0.00

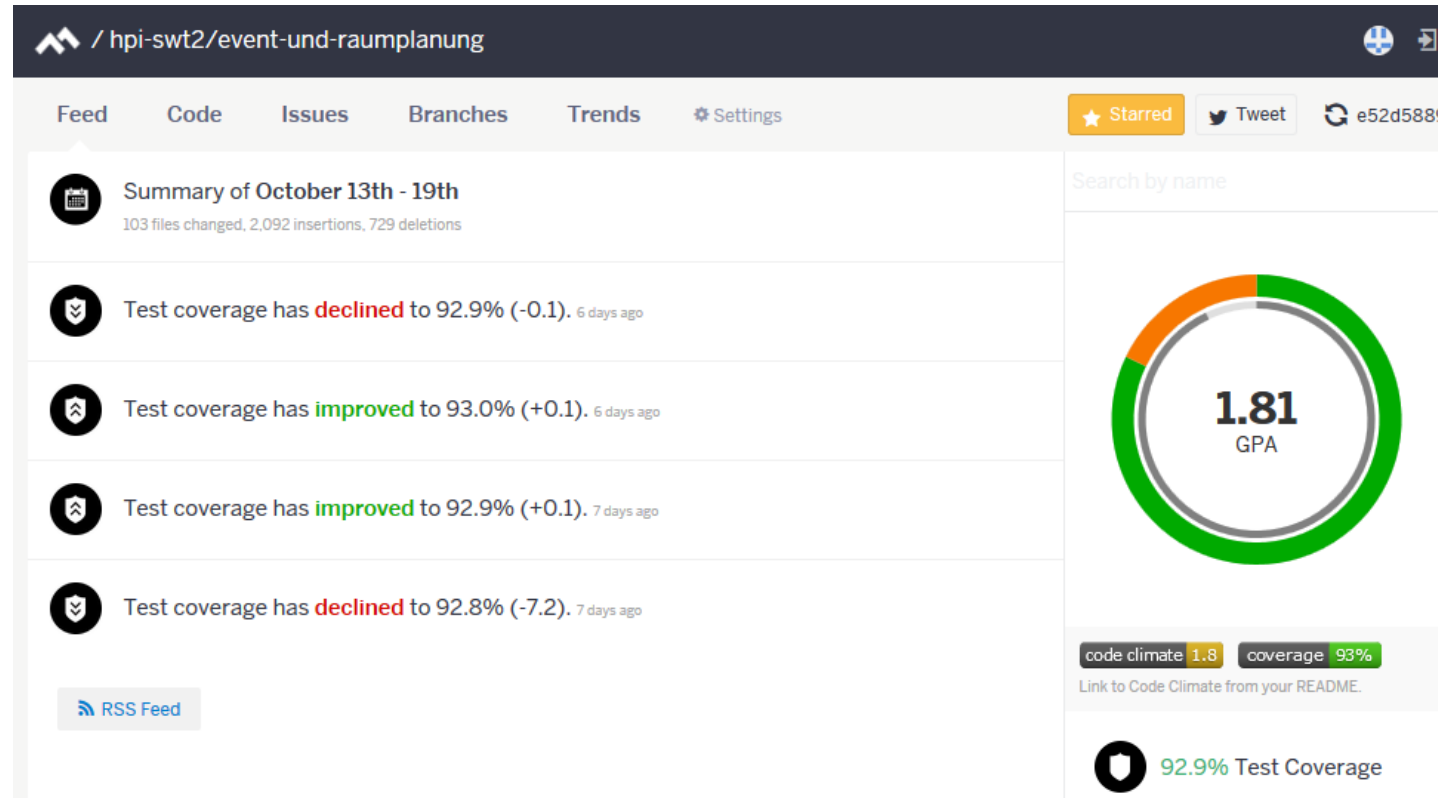
heroku.com/home Blog Careers Privacy Policy Feedback

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



- <https://codeclimate.com/github/hpi-swt2/event-und-raumplanung>



The screenshot displays the CodeClimate interface for the repository `hpi-swt2/event-und-raumplanung`. The top navigation bar includes tabs for Feed, Code, Issues, Branches, Trends, and Settings. The main feed shows a summary of changes for October 13th - 19th (103 files changed, 2,092 insertions, 729 deletions) and a series of test coverage updates: a decline to 92.9% (-0.1) 6 days ago, an improvement to 93.0% (+0.1) 6 days ago, an improvement to 92.9% (+0.1) 7 days ago, and a decline to 92.8% (-7.2) 7 days ago. On the right, a circular GPA badge shows a score of 1.81. Below this, there are badges for 'code climate 1.8' and 'coverage 93%', along with a link to add CodeClimate to the README. At the bottom right, a '92.9% Test Coverage' badge is visible.

All Files (100.0%)   Controllers (100.0%)   Models (100.0%)   Mailers (100.0%)   Helpers (100.0%)   Libraries (100.0%)   Plugins (100.0%)

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

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

Search:

File	% covered	Lines	Relevant Lines	Lines covered
🔍 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

Showing 1 to 6 of 6 entries

Now Get to Work!

# 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](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](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](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](http://commons.wikimedia.org/wiki/File:Utility_pole_in_Curitiba.JPG#mediaviewer/File:Utility_pole_in_Curitiba.JPG)