Introduction & Organization

Scalable Software Engineering
Winter Term 2022/23

Enterprise Platform and Integration Concepts

Introduction: Hello! 👋

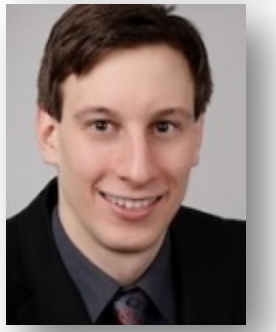
Welcome to Scalable Software Engineering!

- **We're excited to have you all participate!**
- We focus on Agile software development in cooperating teams
- All details/slides on website & Moodle (we'll update with new info):
  https://hpi.de/plattner/teaching/winter-term-2022-23/scalable-software-engineering.html

This is a practical project course

Introduction: Participants






Teaching Team

- Michael Perscheid (michael.perscheid@hpi.de)
- Christoph Matthies (christoph.matthies@hpi.de)
- Ralf Teusner (ralf.teusner@hpi.de)
- Lukas Böhme (lukas.boehme@hpi.de)

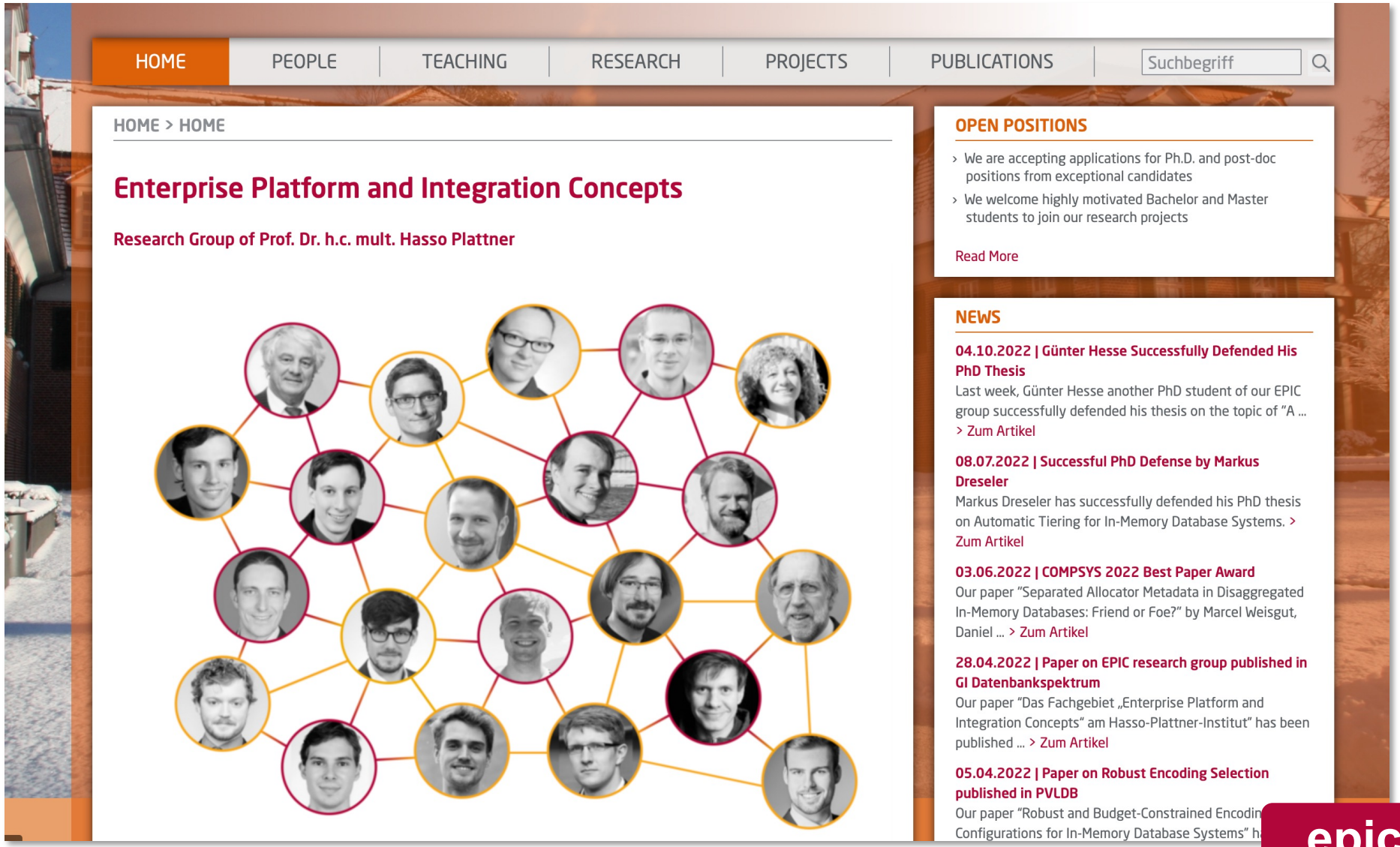
Tutors

- Jost Götte, Paula Klinke, Franziska Hradilak, Nikkel Mollenhauer

Students

- **You!** Without your participation this course won't work!

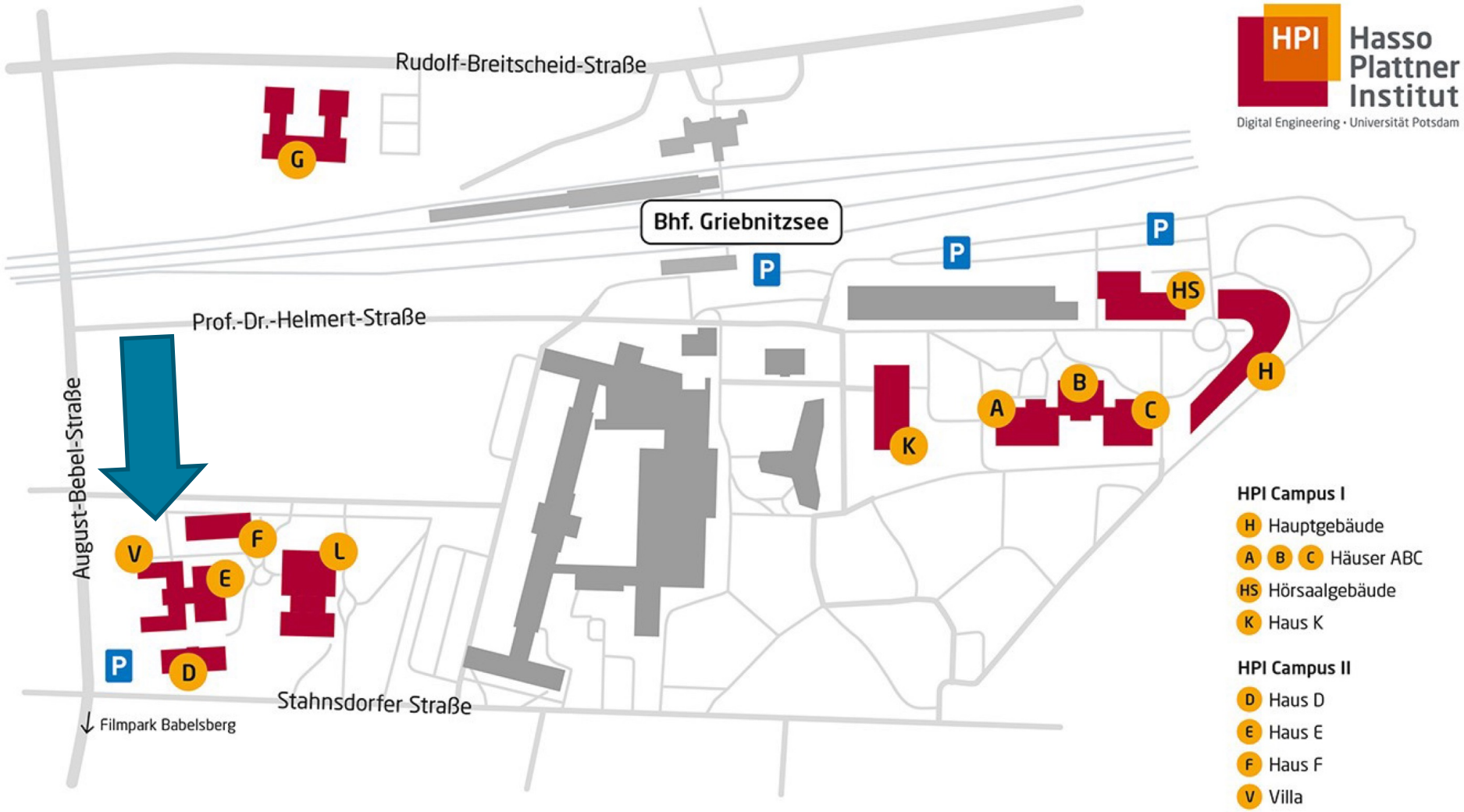
Introduction: EPIC Chair



We also have a Twitter account: @HPI_EPIC

epic.hpi.de

Introduction: EPIC Chair Location



HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

HPI Campus I
- H Hauptgebäude
- A B C Häuser ABC
- HS Hörsaalgebäude
- K Haus K

HPI Campus II
- D Haus D
- E Haus E
- F Haus F
- V Villa
- L Haus L

HPI Campus III
- G Haus G

Introduction: You

- 3rd semester? 5th semester? 1st semester?! Not HPI?
- What are your **previous experiences**...
  - concerning software development team work?
  - concerning web development?

- What are your **expectations for this course**?
  - What do you hope to learn & experience?
  - What is your personal goal?

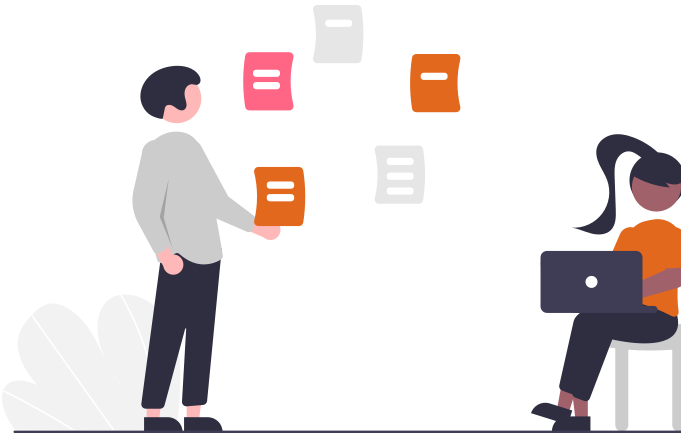
Course Admin

Prerequisites

- Software Architecture and Software Engineering I are highly recommended
- **Interest in learning and working in project teams**

Course

- 4 SWS (≈8h work per week **including lectures**)
  - Some lecture slots will be used for more group work time
  - 6 ECTS credit points (graded)



Grading

Grading & Exams

The final grading is determined by

- Two thirds (⅔): *written exam*
- One third (⅓): p*roject work (project results, project presentations, contributions of teams, application of development process)*

Grading Process

- Grade not solely based on exams
- We want to value the work you put into the project
- **The biggest take-home learnings take place during teamwork**

Course Setup

Scalable Software Engineering

What is Scalable Software Engineering?

- This is a **project course**
- Regular lectures, but focus is on practical work in teams
- You will learn through **experimentation and trying out** collaboration techniques
- **Team meetings are vital**, must make time for them

Learning Objectives

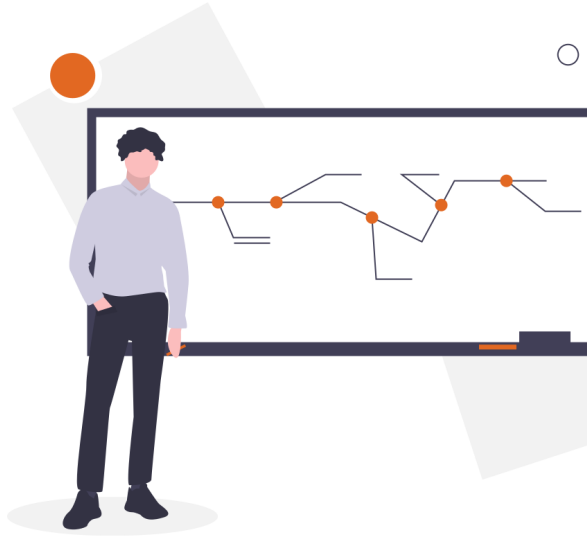
After this course, all students should have

1. Practical **experience with the Agile methods Scrum & Kanban** and their core ideas, artifacts, roles and meetings
2. A working knowledge of **project management techniques** and their practical application
3. Learned how to scale modern software development methods over **multiple collaborating teams**
4. The ability to use **modern development practices**, such as BDD, TDD, CI/CD & DevOps, where appropriate
5. Confidence in using the full feature set of a **source code management (SCM)** and related systems
6. Experienced the value of **rapid release cycles** and continuous integration
7. Learned to critically **self-assess** their role in a team and work towards **collaborative improvement**

What is Scalable Software Engineering?

Lecture

- Scrum and Agile practices in large teams
- Requirements management
- Behavior-Driven-Development
- Project Management
- Development tools
- Agile methods beyond Scrum
- Continuous Integration
- DevOps
- Guest lectures

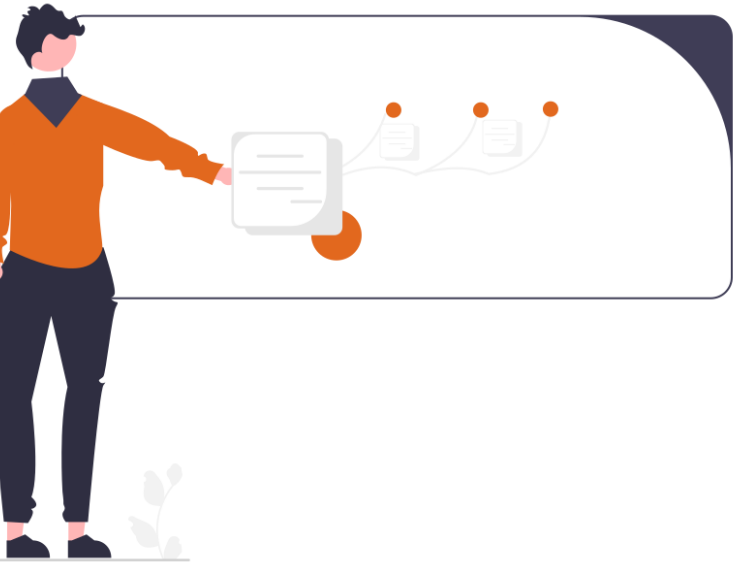


Course Organization

In-Person vs. Remote

- Interactivity and discussions easier in in-person setting
- Lectures will be held in-person
- If necessary, later lectures might be completely virtual
- No hybrid *lecture* setup, teamwork hybrid possible

We encourage masking during the lecture



Course Organization

Collaborative software development

- **You manage, organize and own the project!**
- Realistic (coordination) challenges and problems
- Web programming framework: *Ruby on Rails (who has used that?)*
- Minimal core will be provided, results open source (MIT) on GitHub

Engineering focus

- Understanding of web (MVC) stack and components
- Integration: i.e. avoiding “patchwork” (UI, workflows, data)
- Maintainability of the code base (tests, quality, etc.)
- Functionality *(not on top of the list for a reason)*

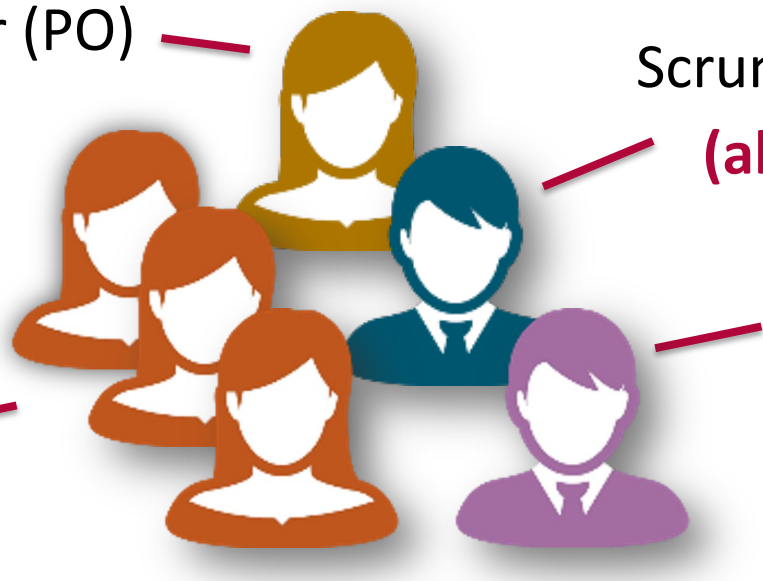
Course Organization

Time management

- **Personal time management is part of the learning experience**
  - Commit to what you can achieve
  - You are responsible to your team and project
- Too much or too little work is a breakdown of planning
  - One fifth of the week
  - Overtime discouraged

Be wary of self-exploitation.
Cooperation is work.

SSE Team Setup



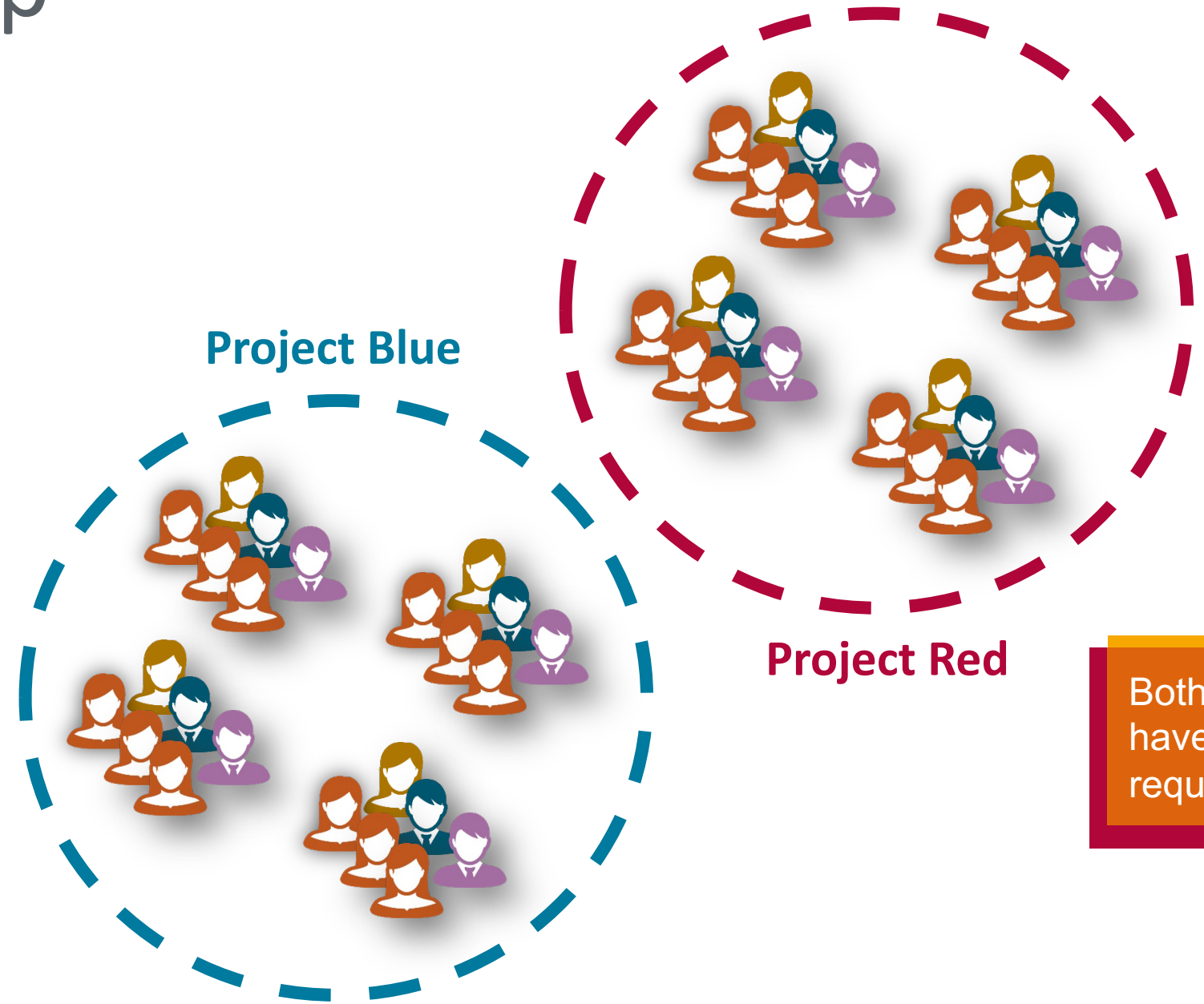
SSE Team Setup

Customer

(has requirements & **represents stakeholders**)

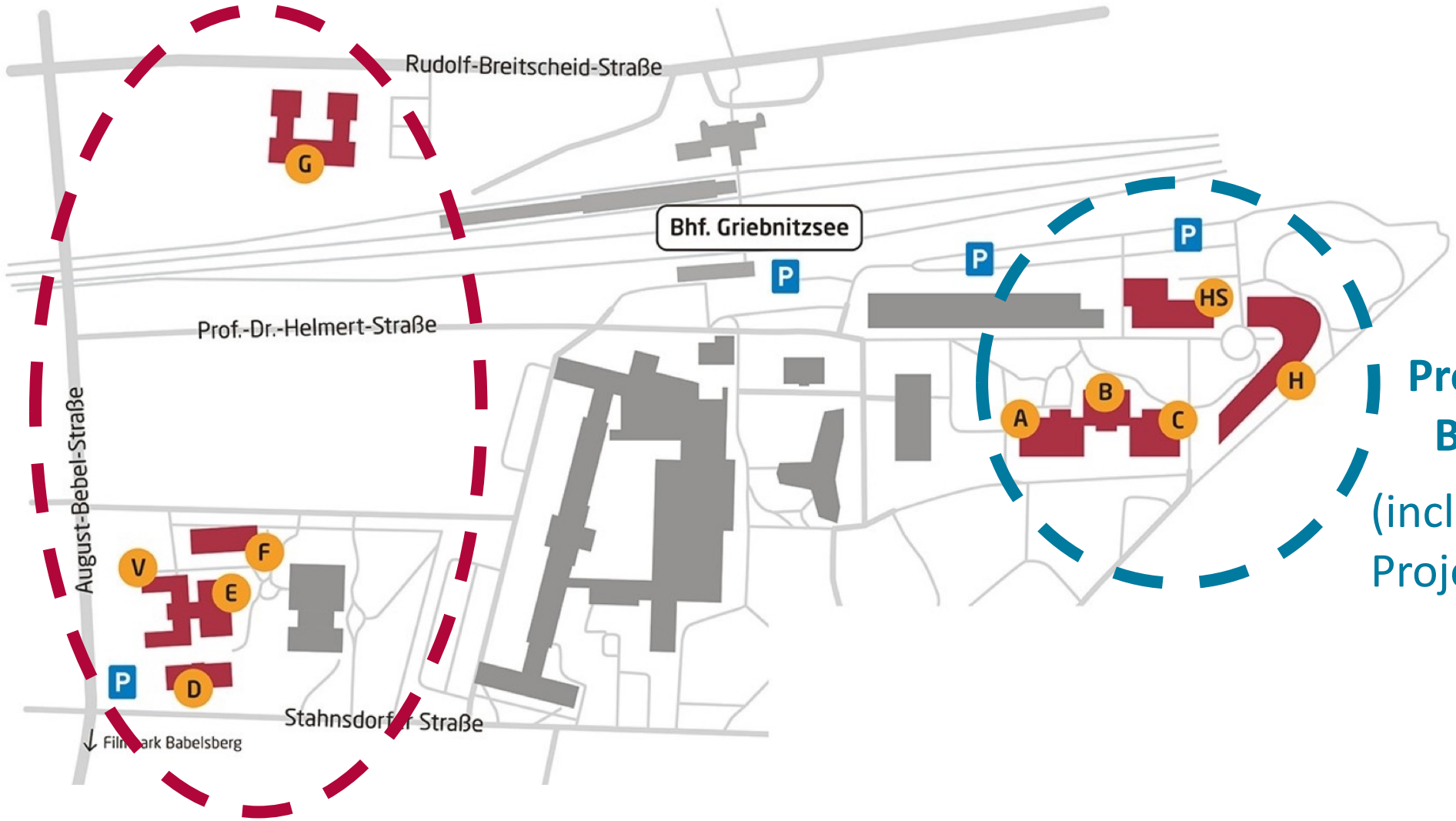
Upper Management

(needs to deliver software to the customer; **hired you & pays you**)



Both projects have the same requirements

SSE Team Setup



Project Red

Project Blue

(incl. Bachelor Project TR1)

SSE Team Setup

Project Teams

- **Project Red**
  - FN
  - HP
  - CL
  - GdM
  - BR
  - BA
- 42 participants

- **Project Blue**
  - MW
  - TF
  - RH
  - AP
  - TR
  - HGHK (HG & ½ HK)
  - PBHK (HG & ½ HK)
- 46 participants

SSE Team Setup

Teams

- Work in your bachelor teams to minimize contacts
- Smaller BP projects fused into larger ones

Teaching Team

- **Tutors as Agile consultants** (coordinate with them!)
  - Present during all major meetings
  - Open for questions, advice & (crazy) ideas
- Lecturers
  - Help with challenges and discussions
  - Workshops with individual roles

Course Schedule

Initial Schedule

- **14.11. Project Kick-Off (Project Vision)**
- Scrum Sprint 1
- Scrum Sprint 2
- Scrum Sprint 3 (Xmas break)
- Kanban phase

- Guest lectures
- Interactive workshops
- January: **Intermediate Presentation**
- Last lecture: **Final Project Presentation**

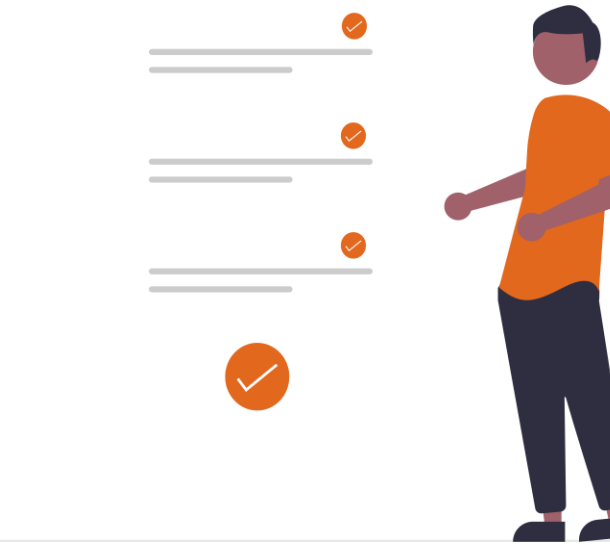
No schedule survives contact with reality

- **Real teamwork brings real challenges**
  - Actually writing a software is vital
  - External constraints may change
- Schedule will adapt
  - Also according to your suggestions

Next Steps

Scalable Software Engineering

Exercise Week 1 (ACTION REQUIRED)



Team Setup

- **Decide** Product Owner, Scrum Master and Project Management **roles**
- Three suggestions for **weekly meeting slot** (virtual or in person?)
  - Exercise slot (Wed 13:30) as a fallback, if nothing else works
- Mail us this info!

Required Reading

- (Re-)read the **Scrum Guide** (https://scrumguides.org)
- Especially the parts on your chosen roles

Until next week (Oct 24th)

Product Owners

- Suggestions for **first (virtual) customer meeting date**
- Mail us this info!

Literature

HPI

Agile literature

- Verner, June M. et al. "In the 25 years since The Mythical Man-Month what have we learned about project management?." *Information and Software Technology* (1999)
- Meyer, Bertrand. *Agile!: The Good, the Hype and the Ugly*. Springer Science & Business Media, 2014.
- Kniberg, Henrik. *Scrum and XP from the Trenches*. Lulu.com, 2015.
- Sutherland, Jeff, and Ken Schwaber. *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game. Scrum.org*.

If you can't find these items in the library or online, please send us an email. We might be able to help.

Bertrand Meyer‘s book is recommeded

Next Lecture

Due to Vollversammung

- Next lecture in exercise slot
- **Wed 13:30**

Mi					Do		
bis KW 48 Modellierungs-sprachen und Formalismen Weske/Seidel ab KW 49 Grundlagen digitaler Systeme Karl HS 1	ÜBUNG Einführung in die Data Science und Machine Learning de Melo HS 3	HCI Project Seminar Personal Fabrication and 3D interaction Baudisch H2.57/58	Internet Security: Weaknesses and Targets Meinel D. Köhler HS 2		Mathematik I – Diskrete Strukturen und Logik Meinel Bethge/Assaf Otholt HS 1	Software-architektur Hirschfeld Lincke/Tauemel Pape/Rein HS 2	Introduction to Image & Video Processing Techniques Trapp Reimann Wattasseril A1,1
Einführung in die Program-miertechnik I Polze Herbrich S. Köhler Wenzel HS 1 + Räume Übung	Mathematik III Stochastik Renard Baum L-E.03				Recht für Ingenieure I Habbe HS 1	Betriebs-systeme I Polze HS 3	(Computer-) Spiele und Theorie Kötzing K-1.02
						Advanced Track Design Thinking (D-School) Termine siehe Web-Seite D-School 09:00 – 14:30 Uhr	
Übung Mod bis KW 48 K-1.02 K-2.03	Übung Software-archi-tektur A1.1 A1.2 A2.1	Übung Datenbank-systeme II Naumann Kaminsky FE.06	Fachspez. Englisch Level 1 Meier G1.E15/16	Übung Skalier-bare Software-entwick-lung Perscheid Teusner Matthies HS 2		Theoretische Informatik I Friedrich HS 1	Complex Event Processing Weske Völker HS 3