Our teaching philosophy

“Tell me and I will forget.
Show me and I will remember.
Involve me and I will understand.
Step back and I will act.”
Interactive Learning

Educators teach and exercise small chunks of content in short cycles. They provide **guidance** to prevent misconception and **feedback** so that learners can reflect on the content and increase their knowledge incrementally.
Our mission

• Make software engineering education great (again)
• Provide realistic examples
• Focus on practical application of knowledge
We need your help!

• 3 courses in the summer

  1) Introduction to Software Engineering (EIST) with 1600 students
  2) Project Organization and Management (POM) with 500 students
  3) Software Engineering Essentials (SEECx) with > 1000 students from all over the world
We search for motivated tutors!

WE WANT YOU!
Participation

1) **Seminar**: 5 ECTS
   - ~ 8 hours per week effort
   - Possible module numbers: IN0014, IN2107

2) **Praktikum**: 10 ECTS
   - ~ 16 hours per week effort
   - Possible module numbers: IN0012, IN2106, IN2175, IN2128

3) **Contract**: between ~ 300€ and ~ 700€ salary per month
   - ~ between 6 and 12 hours per week effort
   - Exact amount of time decided on individual basis
Learning goals

• Learn best practices for teaching
• Organize and conduct exercises and/or classes
• Help students to learn the theory and apply the knowledge in exercises
• Create, conduct and correct exercises
• Deepen your knowledge in software engineering and project management
• Explain concepts to students
• Compare different learning approaches
• Meet new people and have a lot of fun :-)

Details about the courses
Introduction to Software Engineering (EIST)

- **Number of students:** 1600 bachelor students (2nd semester)
- **Lecture:** Thursday, 8:15 - 11:00 (HS1 live and streamed into MW2001)
- **Exercises:** 2h tutor groups per week
- **Exercise concept:** weekly graded homework and project work
- **Tutor responsibilities:**
  - Attend the weekly tutor meeting
  - Prepare and hold 1 or 2 weekly tutor group(s), depending on your participation
  - Correct exercise solutions of students and provide timely feedback to students
  - Help in the final and repeat exam correction
  - Answer student questions (personal and on Slack)
- **Additional responsibilities for ambitious tutors:**
  - Help in the live stream between HS1 and MW2001
  - Help in the review of exercise sheets and provide feedback to the instructors
EIST: Impressions
EIST: Impressions
Project Organization and Management (POM)

- **Number of students**: 500 students
- **Classes (lecture and exercise mixed)**: Wednesday, 8:45 - 11:30 and Friday, 12:15 - 13:45 (HS1)
- **Exercise concept**: Quizzes, individual exercises (in class), team exercises in project work (homework)
- **Tutor responsibilities**:
  - Attend the weekly tutor meeting
  - Attend classes and help students in their individual and team exercises
  - Correct exercise solutions of students and provide timely feedback to students
  - Play the role of the agile product owner in the project work
  - Help in the final and repeat exam correction
  - Answer student questions (personal and on Slack)
- **Additional responsibilities for ambitious tutors**:
  - Hold a tutor lecture about estimation and planning
  - Help in the review of exercises and provide feedback to lectures and exercises
POM: Project team work
Software Engineering Essentials (SEECx)

- **Number of students:** >1000 students from all over the world on the edX platform, ~10 TUM students (bridge course)

- **Lectures:** Online videos

- **Exercises:** Quizzes, programming exercises, peer reviews, project work

- **Tutor responsibilities:**
  - Attend the weekly tutor meeting
  - Correct project work exercises
  - Review the course materials for consistency and difficulty —> propose improvements
  - Correct project work of TUM students and provide timely feedback
  - Help in the conduction of the final assessment

- **Additional responsibilities for ambitious tutors:**
  - Create tutorials as online videos
Example: Quiz with checkmarks

What are the benefits of using delegation instead of inheritance?

1.0 point possible (graded)

☑ Classes do not contain any unneeded methods inherited from the superclass

☐ It is more efficient because objects are encapsulated

☑ It exposes only the required methods of the reused class

☐ It is easier to add new functionality to the reused class
Example: Quiz with drag and drop exercise

Below you will find an incomplete UML class diagram for the description of an antipattern. Drag and drop the elements on their right position.
Example: Automated programming exercise

https://artemis.ase.in.tum.de
Prerequisites for the course - What we expect from you

- You have passed EIST or POM (or a similar course before)
- You have basic knowledge in software engineering or project management
- You are highly motivated
- You support the students in the best way
- You can explain important concepts in an easy manner
Application

1) Fill out the application on
   [https://www1.in.tum.de/lehrstuhl_1/teaching/summer-2018/965-interactive-learning](https://www1.in.tum.de/lehrstuhl_1/teaching/summer-2018/965-interactive-learning)

2) Come prepared to an interview
   (fill out the doodle in the confirmation email)

3) Prioritize in the matching system
   (between: February 9 - 14)

4) We inform you about your participation
Seminar/Praktikum Interactive Learning
Stephan Krusche, Chair for Applied Software Engineering

Info Meeting
25 January 2018