Human Aspects of Software Engineering

Thomas Fritz
Course Objectives

- Broaden your knowledge of Software Engineering by **reading** current and classic literature
- Deepen your understanding of Software Engineering through **reflection** and **discussion**
- Experience a glimpse of Software Engineering through **project** work
- Focus on **research**
Non-Traditional Course Format

Outside of class

■ 2-3 assigned papers per week
  read them, think about them, write a short response
■ small project on research topic
  find a research question, write and present a proposal, do it, write it up and present it

In class

■ Weekly moderator leads a discussion
■ Everyone participates in discussion

No exams! [6 ECTS]
Grading

- Readings, including class participation and 6 response papers [15%]
- Written critiques (review) of a total of two papers [15%]
- Leading discussions [15%]
- Project [40%]
- Peer evaluation of two project reports [15%]
Response Papers

- Encouragement to read and reflect
  Class discussions work better if everyone has read and thought about the paper

- At most half a page to a page per class

- Not a summary. Think of it this way
  If I asked you what you thought about a movie you recently went to, you wouldn’t just summarize it

- Grading based on “thoughtfulness”

- Due by 8pm on Sunday before class
  Submit on: https://www.easychair.org/conferences/?conf=hase13
Response Papers

- Multiple kinds (I will state which one)
  - What did you think about it and what did you find important or interesting?
  - What are main contributions of the paper?
  - What are strengths or weaknesses of the paper/research?
  - What are five questions you have about it?
  - What could be improved?
  - How could you imagine extending the work?
  - Do you agree or disagree with the findings?
  - …

- Express your perspective, address all readings and draw connections between readings when possible
Discussions

- Discuss the research
- Share your opinions, ideas and thoughts
- Ask questions about the work
- See what others thought
- Listen and speak actively
- Look for value not just flaws in reading
Written Critiques of two papers

- Research communities rely on peer reviews of results, if you want to be a researcher you need to learn about critiquing research papers

- One for first half of the course (first 7 weeks), one for second half

- Choose two topics and I will assign papers

- Templates will be provided
  
  Summary, what are its strengths, what are the weaknesses
  
  (realize that the authors would usually read it, so be constructive)

- No response paper needed that week!
Leading Discussions

- For about 3 to 4 papers (depending on class size)

- Prepare for discussion
  - Prepare brief introduction (less than 5 minutes)
  - Provide additional background on subject
  - Prepare interesting and challenging points and questions for discussion
  - Think about overall structure, how to make it interactive and how to keep discussion going

- Hand in a brief (one paragraph) description of your plan

- Sign-up will be online

- No response paper needed that week!
Moderation

- Introduce and provide background
- Moderate an interactive discussion and include other student’s perspectives and ideas
- Involve all participants
- Give impulses for discussion
- If an interesting topic comes up, be flexible about veering from your discussion plan
- Be respectful
Project

- Gain depth in one area of the course, four options:
  - Build a software tool to solve a problem developers encounter
  - Literature survey
  - Evaluate an advanced software development tool
  - Study of existing practice

- Each accompanied by a paper (max. 5 or 10 pages)

- Individual or in groups (up to 3 people, depending on class size)

- One page project proposal due on **October 9th**

- Project presentations at end (10 minutes)
Project Continued

- Project report [ACM paper format]
  - Build: describe too, sample use and comparison to existing tools; (no more than) 5 pages
  - Survey: describe and critique papers to provide state-of-the-art report on topic; (no more than) 10 pages
  - Use: describe your experience, including critique of technique/tool; (no more than) 5 pages
  - Study: describe the results, show potential for future research and compare to existing work

- You will have your project approved by me

- One-on-one meetings shortly after project proposal is due
Build a Software Tool

- Experience issues in building tools to support software development

- Identify real problem developers face and model solution (draw upon your own experience)

- Implement the tool, might be:
  - an implementation of an algorithm proposed in a paper we read
  - visualization of information produced by existing tool
  - new idea you have for a tool

- Evaluate the tool (preliminary)
  - probably qualitative scenarios (proof-of-concept) to show it works
State-of-the-Art Literature Survey

- Gain a more in-depth understanding of a topic you find interesting in class
- Find and read additional research papers on the topic
- Write a critical summary of the papers describing the current state-of-the-art for that topic
- Synthesis is key: what are relevant papers, how do they contribute to understanding, what shortcomings do they identify, how can future work address these
- This option will be evaluated most rigorously
Use and Evaluate Existing Tool

- Gain experience using one or more advanced software development technique or tools
- Use a software development technique or tool(s) that is relevant to a topic in class and apply it to a problem/system you choose
Study of Existing Practice

- Investigate a specific aspect of software development practice
- Design and conduct interviews or survey with developers
- Apply open coding to the collected (and transcribed) answers
- Summarize results and show how the results open potential for future research
Peer Evaluations

- Assess projects like a program committee
  - Everyone will read and review two project reports
  - Reviews are organized via easychair
    http://easychair.org

- Hand in review (will also be sent to authors)

- Program committee meeting in the last class
  - up to you whether we ‘accept’ papers
  - acceptance has no bearing on grade
The professor and the TA

- We’re here to help
- Talk to us if you want feedback or need help
- Talk to us if you do not find a topic or want to discuss your idea
- In class, we’re here to discuss
Introduce yourself!
Focus of Course

- Individual development and development activities (rather than processes, planning, requirements, ..)
- Studies of programmers
- Tools to support software developers
- Evaluation of these tools with developers
Topics

- **Software Evolution**
  - Refactoring, reverse engineering, design

- **Program Comprehension**
  - Information needs, code navigation, code search

- **Development Team Process**
  - Global software development

- **Tools and Environments**
  - Awareness, debugging, task-centric development, visualization

- **Quantitative & qualitative SE research**
  - Experiments, case studies,…
Topic list

- Historical Context (next week)
  - Brooks. *No Silver Bullet*.
  - Gibbs. *Software's Chronic Crisis*.

- Global Software Engineering
  - Bird, C. et al. *Does distributed development affect software quality?*
Topic list

- Design
  - Gail C. Murphy et al. *Software Reflexion Models: Bridging the Gap between Design and Implementation.*

- Developer's Information Needs
  - Jonathan Sillito et al. *Asking and Answering Questions during a Programming Change Task.*
Topic List

- **Supporting Developer Queries**
  - Michael Würsch et al. *Supporting developers with natural language queries.*
  - D. Janzen et al. *Navigating and querying code without getting lost.*

- **Tools**
  - Vineet Sinha et al. *Relo: Helping Users Manage Context during Interactive Exploratory Visualization of Large Codebases*
  - Robert DeLine et al. *Software Development with Code Maps*
Topic List

- Code changes
- Awareness
- Naturalness and Language
- Using developer interactions
- Psycho-physiological aspects (mood, eye-tracking)
What would you like to learn about?

Any topics you are missing in the list?
My research interests...

- Tackling Information Overload
  - Information Fragments, Degree-of-Knowledge
- Code/Information Navigation
  - context for searches, recommendations
- Personal Analytics
  - provide retrospective analysis
- Human Sensing
  - psycho-physiological features to measure task difficulty
- ...

To Dos

- Choose 2 papers you would like to present in first 7 weeks
  - Insert into Google doc (by Sept 23 @ 8am); first come first serve

- Start thinking about projects

- Get an easychair account (free)
Next week

- Two high-level papers for next week

- Read and write a short (half page) response paper

- Submit on easychair: [https://www.easychair.org/conferences/?conf=hase13](https://www.easychair.org/conferences/?conf=hase13)
More Information

- See website:
  [http://www.ifi.uzh.ch/seal/teaching/courses/hase.html](http://www.ifi.uzh.ch/seal/teaching/courses/hase.html)

- Contact:
  - Thomas Fritz
    fritz@ifi.uzh.ch
  - Katja Kevic
    kevic@ifi.uzh.ch