Software Development and Evolution

Seminar in Advanced Software
Engineering, FS 2014

Thomas Fritz

s.e.a.l.
software evolution & architecture lab

Organizational Announcements

- 3rd year and up
  (prerequisite: Software Engineering)

- Language for report and presentation is English

- Work independently

- Don't forget to do the "Modulbuchung"!

Seminar Goals

- Introduce you to a part of research in software engineering

- Improve your ability to critically read and analyze scientific papers

- Strengthen your technical writing and your presentation skills

- Improve your ability to ask and answer critical questions

What you have to do

- Do a literature review for an assigned topic
  (at least 10 relevant articles)

- Write a report and refine it
  (12 to 15 pages, LNCS format)

- Review two reports

- Present your findings
  (approx. 15mins + 10mins questions)

- Actively participate in discussions

3 credits / 90 hours: approx. 70 report & review, 20 presentation

Software Development and Evolution - Topics

- Supporting Crosscutting Concerns

- Code Recommendation

- Concept Location and Code Search

- Tool Support for Tasks/Bugs

- Software Quality Tools

- Refactoring

- Software Architecture

- APIs

- Human Aspects in Software Development

Topic – Supporting Crosscutting Concerns

- **Aspect-Oriented Programming.**

- **N degrees of separation: multi-dimensional separation of concerns.**

- Concern graphs: finding and describing concerns using structural program dependencies.

Topic – Code Recommendation

- Using structural context to recommend source code examples.

- Mining version histories to guide software changes.

Topic – Concept Location and Code Search

- Feature identification: where in the code is this feature implemented.

- Where should the bugs be fixed? Using information retrieval and more to localize bugs.

Topic – Tool Support for Tasks/Bugs

- Using task context to improve programmer productivity.

- Who should fix this bug?

- Summarizing software artifacts: a case study of bug reports.

Topic – Software Quality Tools

- Yesterday my program worked today it does not. Why?

- Change analysis with Evolizer and ChangeDistiller.

- Tools for continuously evaluating distributed-system qualities.

Topic – Refactoring

- A survey of software refactoring.

- How we refactor and how we know it.

- Automated testing of refactoring engines.

Topic – Software Architecture

- Using software evolution to focus architectural recovery.

- ArchJava: connecting software architecture to implementation.

- Software Reflexion Models: bridging the gap between source and high-level models.

Topic – APIs

- What makes APIs hard to learn? Answers from developers.

- How to design a good API and why it matters.

Topic – Human Aspects in Software Development

- Using fMRI (functional magnetic resonance imaging) to examine program comprehension

- Using eye-tracking for characterizing program comprehension

Finding Relevant Work

- search online by author, keyword, topic, etc. on personal web sites, Google Scholar, ACM Digital library, Citeseer, IEEE Digital Library

- Look through proceedings of main conferences (ICSE, FSE, ESEM, ICGSE, ICPC, CSCW)

- Browse and follow references/citations in relevant papers and read related work sections

- If you found a relevant and older paper, look for papers it is "cited by"

Code bubbles: rethinking the user interface paradigm of integrated development environments

Full Text: 📄 PDF

Authors: Andrew Bragdon   Brown University
            Steven P. Reiss   Brown University
            Robert Zeleznik   Brown University
            Suman Karumuri   Brown University
            William Cheung   Brown University
            Joshua Kaplan   Brown University
            Christopher Coleman   Brown University
            Ferdi Adeputra   Brown University
            Joseph J. LaViola, Jr. University of Central Florida

Published in:

            · Proceeding
            ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering -
            Volume 1
            ACM New York, NY, USA ©2010
            table of contents ISBN: 978-1-60558-719-6   doi>10.1145/1806799.1806866

2010 Article

Bibliometrics

· Downloads (6 Weeks): 13
· Downloads (12 Months): 258
· Citation Count: 11

🎙 **Feedback** | Switch to single page view (no tabs)

Abstract	Authors	References	**Cited By**	Index Terms	Publication	Reviews	Comments	Table of Contents

11 Citations

◆ Robert DeLine , Gina Venolia , Kael Rowan, Software development with code maps, Communications of the ACM, v.53 n.8, August 2010

◆ Robert DeLine , Gina Venolia , Kael Rowan, Software Development with Code Maps, Queue, v.8 n.7, July 2010

◆ John Hardy , Christopher Bull , Gerald Kotonya , Jon Whittle, Digitally annexing desk space for software development: NIER track, Proceeding of the 33rd international conference on Software engineering, May 21-28, 2011, Waikiki, Honolulu, HI, USA

16

Reading a Research Paper

- Read *critically*: be suspicious and ask appropriate questions:

  - e.g. are the authors solving the right problem, what are the limitations, are the assumptions reasonable

- Read *creatively*: critical is easy, reading creatively is harder:

  - e.g. what are the good ideas, how would you extend it, are there applications or extensions the authors haven't thought of

- Make *notes*!

- After reading, try to *summarize* the paper

- *Compare* to other works

Expectations to Report

- Summarize current state of the art
- Provide a good overview of the area
- Present main research questions, concepts, ideas and approaches in the area as well as open challenges
- Find commonalities, specialties, differences,…
- Critical and creative thinking, some reflection on your own

- Wikipedia is not an option!

Expectations to Report (2)

- Find good structure / outline / categorization and present in a coherent and consistent way

  Abstract, Introduction, Related Work, Discussion, Conclusion, References, Word of Honor

- Use **correct and understandable English**, presentation is very important (proof-read?)

- Phrases such as "I like this paper" should not be in it

- Cite and quote correctly to avoid plagiarism!

- Higher for master than for bachelor students

Find more details at (Sven Seuken):
http://www.ifi.uzh.ch/ce/teaching/fall2013/seminar/seminar_guideline.pdf

Word of Honor

- At the end of your report, include a note on a separate page which you sign, stating:

  I, [first and last name], hereby declare that I have produced this work independently and have used no other than the listed tools and sources

- This does not count towards the 12-15 pages
- Only required for the final report!

Review a Report

- Start with a brief summary of the report (2-3 sentences)

- Technical quality, originality/novelty and significance:
  are the arguments in the paper correct, how original/novel is the report, how significant is the research question the author poses, is the research area well covered, what is good about the report, are there any problems/issues, what could the author improve

- Logical structure, presentation and style:
  is the paper coherent, well-written and are concepts and approaches well-explained, are graphics/tables used appropriately, is it easy to follow and clear, how could it be improved

- Be constructive, polite and professional!

- Start with summary, pros/cons and go from high granularity to lower

Review a Report

- You will receive a review form through EasyChair

- Provide your review and a grade on a scale from 0 to 6 (minimum increments of 0.25)

  5 - 6: An excellent work.

  4 - 5: A good work with just a couple of small weaknesses.

  3 - 4: An average work with clear weaknesses.

  0 - 3: Insufficient work with many substantial weaknesses.

Presentation (15 mins)

- Several guidelines on website
- Some more:
  - Don't exceed the time limit!!!
  - Practice the talk
  - Don't ignore the audience
  - Avoid too many slides, too many bullets, fonts too small, too much text
  - Have a flow
  - Motivate topic, explain concepts, provide overview,…

- Scientifically Speaking: http://www.erp.wisc.edu/profdev/Scientifically_speaking.pdf

Deadlines

- Mar 6th –   Email with 3 preferences to me (latest)

- Mar 7th –   Topic assignment

- Apr 6th –   Report submission (midnight hard deadline)

- Apr 16th – Review period end

- Apr 17th – Notification

- Apr 27th – Corrected report submission

- May 9th – Presentation Day (9am, mandatory)

More Information

- See website:
  http://www.ifi.uzh.ch/seal/teaching/courses/seminar.html

  (also includes presentation guidelines)

- Contact:

  Thomas Fritz   fritz@ifi.uzh.ch

BSc and MSc Project…

- Answering developers questions
- Biometric Sensors in SE
- Personal Analytics
- …

…contact me if you are interested