Free for All: Open RE Course - Lectures on YouTube, Slides on Slideshare & Material on Foss2Serve

Birgit Penzenstadler[a,b]

[a] Chalmers, Sweden
[b] LUT, Finland

Abstract

In the endeavour of making RE Education more accessible and available, I shared my semester-long Master-level course materials online. The videos are available on YouTube and have received 25k views from around the world, and the slides, assignments, and suggested readings are provided on Foss2Serve, a portal developed under NSF funding. The objective of this short paper is to explain what the available resources are, how this resource has come together and how it has been used. All materials are under Creative Commons Attribution-ShareAlike 4.0 International License and have been used at universities in at least 5 countries plus by individuals globally.

Keywords

requirements engineering, education, open source, open science

1. Introduction

Many of us develop new course materials or rework our course materials every year, and it is a lot of work - which maybe we could streamline more if we crowd-sourced. Hence my offer. I have been iterating my materials for teaching Requirements Engineering starting at TUM[1], then UCI[2], and then CSULB[3]. In 2016/2017, two events prompted me to make my teaching materials accessible and to share them via Creative Commons. First, I got involved with the Humanitarian Free Open Source Software community [1] and decided to include that aspect in my teaching. Second, faculty at CSULB were encouraged to plan towards holding hybrid courses as we were short of classrooms, which led me to explore the support for online teaching materials and the Learning Glass[4], a whiteboard simulator for distance learning. I embarked on the scary journey of offering all my teaching materials to the scrutinizing eyes of the Internet.

In: F.B. Aydemir, C. Gralha, S. Abualhaija, T. Breaux, M. Daneva, N. Ernst, A. Ferrari, X. Franch, S. Ghanavati, E. Groen, R. Guizzardi, J. Guo, A. Herrmann, J. Horkoff, P. Mennig, E. Paja, A. Perini, N. Seyff, A. Susi, A. Vogelsang (eds.): Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12-04-2021

birgitp@chalmers.se (B. Penzenstadler)
https://birgit.penzenstadler.de/ (B. Penzenstadler)
0000-0002-5771-0455 (B. Penzenstadler)

[1] Technische Universität München, Germany: http://www.tum.de
[2] University of California, Irvine: http://www.uci.edu
[3] California State University Long Beach, CA, USA: http://www.csulb.edu
[4] https://www.csulb.edu/academic-technology-services/media-services/learning-glass

Since the release of the materials under Creative Commons Attribution-ShareAlike 4.0 International License[5], they have been used by colleagues to support their own materials, by peers I have never met but who reached out (and maybe some more who haven't) and by students to supplement whichever learning materials they were given. Furthermore I have received a number of messages by practitioners who used the materials to brush up on certain aspects or to expand their knowledge. From enquiries that I received I know that the materials have been used in at least 5 countries on three continents (Europe, North America, South America).

The course is by no means perfect, but it is a solid foundation to build upon and free for all to use and adapt.

2. The Course: Outline & Materials

The graduate course on Requirements Engineering was held every Spring semester in a fifteen week course that meets twice per week. Each meeting consists of a 50 minute lecture and a 75 minute lab session. The course follows the thematic outline presented in the left column of Table 1. The assignments that students carry out in the lab are composed of individual and team assignments. The team assignments are parts of a requirements specification, so that over the course of the semester, each team develops a complete requirements specification for a system.

In previous years, students were allowed to come up with their own system ideas to develop specifications for. While that sparked the creativity of some students, a more likely scenario they will be developing specifications for in the future is that there are legacy systems that need to be analyzed and adapted or extended. One solution to this is using an HFOSS system to analyze and evolve — at the same time, many HFOSS systems do not have well documented requirements and therefore this is a potential win-win for both the open source community (because the students contribute) and the students (because they get to work with a real-life system including adequate complexity and challenges). For the HFOSS integration, we adapted the assignments to reengineer specific requirements artifacts as opposed to "inventing" requirements for them.

All slides are available on slideshare [2], and the syllabus including breakdown of the structure, learning goals, assignments, and suggested readings are available on the FOSS2serve wiki [3]. The assignments can be used as is or reverted back to being used with a self-selected software system, either an existing one to be evolved or a new one to be developed for the first time.

The YouTube playlist [4] is the full course in Requirements Engineering as I have held it for several years at CSULB. The numbered lectures are recordings from the Learning Glass. The other videos use presentation slides to give more details and background on the contents touched upon in the learning glass recordings. The videos are based on the same content outline, and so they can be used for flipped learning, where the students watch the Learning Glass lecture at home and then the content can be discussed in class with support of the slides [2][6].

[5] https://creativecommons.org/licenses/by-sa/4.0/

[6] Additionally available in a reusable and modifiable format at https://github.com/reet-workshop/activities

Table 1
Course content and assignments of Requirements Engineering with HFOSS

W	Lectures	Labs / Activities
1	**Course Introduction** * Why do we need Requirements Engineering and what is it? * Introduction to natural language requirements	**Open source intro and EARS** * Brief overview of HFOSS and the planned activities * Review HFOSS projects for interest (http://openmrs.org) * Practice EARS (Activity) approach for OpenMRS requirements
2	**Processes and frameworks** * What is the process and what are the roles that perform RE? * What reference structures and standards can I use for requirements?	**Requirements engineering process (Activity)** * Identify the RE process steps used in HFOSS project * Compare to traditional RE phases and artifact-oriented RE. **Mapping Requirements Specification Standards (Activity)** * Research the ISO RE standard 29148 and requirements specification templates.
3	**Artifact-oriented requirements engineering** * Artifact model for domain-independent RE (AMDiRE) * Why are we building this system?	**Exploring artifact models (Activity)** * Find the requirements documented for OpenMRS and sort the information into a requirements specification template.
4	**Stakeholders** * Who are the people to talk to about requirements?	**(Re-)Engineering stakeholders (Activity)** * Make list of stakeholders in HFOSS project * Make stakeholder model (deliverable)
5	**Goals** * What are the major objectives for the system? * How to differentiate the different types?	**(Re-)Engineering goals (Activity)** * Make list of goals in HFOSS project * Make goal model (deliverable)
6	**System Vision** * What exactly do we want to achieve? * What information sources do I need?	**(Re-)Engineering a system vision (Activity)** * Making a rich picture for the future OpenMRS
7	**Domain Models** * What are the surrounding systems ours interacts with?	**(Re-)Engineering a domain model (Activity)** * Develop a UML domain model for a system under analysis.
8	**Midterm** * Recap, questions and answers * Midterm solution	**Q & A session** * Discussion
9	**Usage Models** * How will the system interact with the user?	**Use cases** * (Re-)Engineering use cases (Activity) * Refine/rework use cases after feedback (deliverable)
10	**Scaling RE and RE tools** * How to adapt RE for a specific project setting? * How to select RE tools?	**Tools and assessment** * Try out session for RE tools * Write an assessment of an RE tool
11	**Non-functional requirements** * How to deal with requirements that are not about functionality? * How to specify which qualities need to be met?	**Non-functional requirements** * (Re-)Engineering Quality requirements (Activity)
12	**Quality models** * How to determine the quality characteristics?	**Quality models** * Perform peer review of non-functional requirements elaborated by other team and give feedback on how to improve them.
13	**Quality assurance** * How to ensure that RE is done in a good way?	**Quality assurance in documentation** * Discuss Quality Assurance Assignment sheet questions
14	**Requirements management** * How to evolve requirements? * How to anticipate and plan for risks. * How to put it all together	**Requirements management** * Bug tracker activity * Requirements Rationales (lab discussion on article)
15	**Research and Recap** * Research topics * Recap of the 2nd half of the semester	**Final exam** * Final practice

3. Related Work and Conclusion

Pinto et al. [5] interviewed professors who had introduced the process of contributing to an existing, non-trivial OSS project as part of a software engineering course. They observed that this process enhances students' technical skills, enforces students to learn social skills, and, as a result, improves students' resumes. Some of the fellow educators who have incorporated this on the graduate level are Jaccheri and Osterlie [6] for action research, Lundell, Persson, and Lings [7] for FOSS projects, Beecham et al. [8] for global software engineering as an alternative to a multi-university approach, Van Deursen et al. [9] for a collaborative software architecture course, and Sarma et al. [10] for specifically the onboarding in projects.

To the best of my knowledge, there is no RE course yet that makes all materials available for use and, in addition, offers an integration with open source content.

4. Discussion

The **structure and contents** of the course are based on ten years of experience in teaching requirements engineering. While the author has not worked in industry full-time during that period, the relation with industry practices is maintained by frequent conversations at conferences and in her private network. Some of the techniques taught in the course have to this day not found wide application in industry (but could incur significant benefits), others are widely used and

The **challenges and lessons learned** on the implementation of the open source version are that projects be may difficult to interact with in a limited course setting unless the instructor is a long-standing contributor well acquainted with the community. The version independent of a specific open source project always let students choose their own project ideas which sometimes led to great results (but no strong relation to industrial practice where greenfield development isn't the norm) and other times to contributions that could synergistically benefit another course.

The **feedback** to report on is four-fold: Alumni who graduated in the mean time, students from other universities and countries who have used the materials in addition to courses or because there was no dedicated course, practitioners who wanted to expand their knowledge, and educators who have used the materials in their own teachings. I have received acknowledgement from alumni for the usefulness of this course in their practice now, from students from abroad who found it useful to complement their education, from practitioners who wanted to expand into business analysis or just broader their understanding, and from instructors who were grateful to be able to reuse existing materials.

The **current limitations** are mainly related to coursework in general: no lecture ever seems perfect and there are always new approaches that could be introduced. That said, this version of the course is a good baseline to start with.

The **opportunities for further improvement** are mainly related to making the content easier to teach online, for example by support of quizzes and reflection questions that would otherwise be discussed in class.

5. Conclusion

My personal objective is to make educational resources freely available wherever I can, to dedicatedly support educators and (aspiring) students in countries or at universities with less resources. I hope the resources continue to be used widely and help out educators, students and interested practitioners alike.

Acknowledgments

I thank Heidi Ellis, Darci Burdge, Stoney Jackson, Lori Postner, Greg Hislop, Gina Likins and Cliff Kussmaul for a great POSSE workshop in Raleigh, NC, on how to better incorporate FOSS into teaching. Part of this work has been made possible by part of the NSF grant DUE-1225688 received as subcontractor to develop some of the assignments used in this course.
Also thanks to the anonymous reviewers for suggesting discussion items.

References

[1] H. J. Ellis, G. W. Hislop, S. Jackson, L. Postner, Team Project Experiences in Humanitarian Free and Open Source Software (HFOSS), ACM Transactions on Computing Education (TOCE) 15 (2015) 18.

[2] B. Penzenstadler, Requirements engineering, 2017. https://www.slideshare.net/kamikitty/requirements-engineering-introduction.

[3] B. Penzenstadler, Requirements engineering, 2017. http://www.foss2serve.org/index.php/Requirements_Engineering,_CSU_Long_Beach,_Penzenstadler.

[4] B. Penzenstadler, Requirements engineering, 2017. https://www.youtube.com/playlist?list=PLUgFMzuE8lQDeixpbP3s6EyQx8PiNdeQL.

[5] G. Pinto, F. Figueira Filho, I. Steinmacher, M. A. Gerosa, Training software engineers using open-source software: the professors? perspective, in: The 30th IEEE Conference on Software Engineering Education and Training, 2017, pp. 1–5.

[6] L. Jaccheri, T. Osterlie, Open source software: A source of possibilities for software engineering education and empirical software engineering, in: 1st Intl. Workshop on FLOSS, IEEE, 2007, pp. 5–5.

[7] B. Lundell, A. Persson, B. Lings, Learning through practical involvement in the oss ecosystem: Experiences from a masters assignment, Open source development, adoption and innovation (2007) 289–294.

[8] Beecham et al., How best to teach global software engineering? educators are divided, IEEE Software 34 (2017) 16–19.

[9] Van Deursen et al., A collaborative approach to teaching software architecture, in: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, ACM, 2017, pp. 591–596.

[10] Sarma et al., Training the future workforce through task curation in an oss ecosystem, in: Proc. SIGSOFT Intl Symp on Foundations of SE, ACM, 2016, pp. 932–935.