Teaching Software Engineering in Primary and Secondary Schools

Peter Antonitsch[1)], Andreas Bollin[1),2)], Stefan Pasterk [1)], Barbara Sabitzer[1)]

[1)] Institute for Informatics and Informatics Didactics, University of Klagenfurt, Austria

[2)] Software Engineering Research Group, University of Klagenfurt, Austria

Abstract

Software is everywhere – be it in mobile phones, in washing machines, or in cars. With it, the importance of Software Engineering (SE) is uncontested, and it is taught all over the world: at Universities, at Colleges, and recently also at High Schools. There are international Software Engineering curricula, standards, and certificates, but there is no manifestation of SE (and related practices) in the course syllabi at primary and secondary schools. Most important, SE is not just programming.

Taking a closer look at SE, its main goal is to develop programs that are affordable and dependable for consumers without bugs or glitches. In order to do so, SE education must account for a broad spectrum of knowledge and skills that software engineers will be required to apply throughout their professional life. Covering all the topics in depth within a school setting (from primary to secondary schools) seems to be infeasible due to the previous knowledge of the pupils, the curricular constraints as well as due to the inherent differences between the school types. Similar arguments hold for the teachers, as most of them are not really trained in SE.

Now, based on the authors' experiences gained in combining SE topics with school projects in a vocational high school for commerce and tourism (11th grade) in cooperation with a lower secondary school (6th grade) it turned out that, by customization of the approach, one is able to address pupils with different maturity levels, educational aims, and backgrounds.

The objective of this 90-minutes workshop is to show that it is possible to interweave SE topics with school projects and to motivate for the most important practices related to that field. Key skills and challenges are identified, mapped to the situation at hand, and, by following a stepwise approach, example settings are discussed.

Biographies of workshop leaders

Peter Antonitsch

Peter K. Antonitsch is doing research work in the field of the didactics of informatics at the University of Klagenfurt/Austria and teaches informatics at a secondary school in Klagenfurt at the same time. Having earned a doctoral title in mathematics from the University in Vienna/Austria, his interest stretches from the formalization of concepts to the methods of teaching informatics, with current research work focusing on how to teach algorithmic thinking at different age levels.

Andreas Bollin

Andreas Bollin received his Ph.D. and Venia Docendi from the University of Klagenfurt where he is currently active as Full Professor at the Informatics Didactics Group. His teaching and research activities focus on various software engineering topics, like software engineering education, project management, and software development processes. As an experiences teacher his as been giving lectures at several European universities for more than 18 years. He is member of the ACM, IEEE, and the OCG.

Stefan Pasterk

Stefan Pasterk holds a master degree as teacher of mathematics and informatics and currently works as project assistant at the Informatics Didactics Group at the University of Klagenfurt. After two years as student assistant, he is now working on his dissertation in the field of informatics didactics. For the project 'Informatics – A child's play' he developed an informatics curriculum as well as several teaching units and materials, which he implemented in different school types and levels in order to provide especially young children an insight into topics of informatics.

Barbara Sabitzer

Barbara Sabitzer is assistant professor at the Department of Informatics Didactics at University of Klagenfurt, lecturer (didactics, educational psychology) at the University College of Teacher Education in Carinthia, author of schoolbooks for several languages and teacher (informatics, foreign languages) in a vocational high school. Her research interests range from informatics didactics and neurodidactics over technology-enhanced learning to cross-curricular teaching with a special focus on computational thinking in general education and language learning.

Materials provided

- The SEEK knowledge areas of Software Engineering Education
- Instructions for creating and editing problems

Agenda of the Workshop (90 Minutes)

Part 1 - Introduction to the Software Engineering Education (30 min.)

Understanding the motivation behind Software Engineering, overview of the SEEK knowledge areas.

Part 2 – Group Work (20 min.)

Depending on the number and skills of the attendees, the group is split into two (up to 4) sub-groups. Each group works out the schedule for a (cross-curricular) school project with Software Engineering topics embedded.

Part 3 – Reflection (20 min.)

Getting an overview of the groups' projects. Discussion of impediments and how to solve them.

Part 4 - Discussion and Summary (20 min.)

Getting feedback to steer future development of Software Engineering in schools.