

Chapter 3

Fostering Practical Developers in Computer Science Classrooms: A PBL Approach

Yin Zhang

Northeastern University, China

ABSTRACT

Real-life software development requires practical developers. This chapter discusses the challenges put by real-life software development on computer science education of modern universities, and how to face these challenges by changing traditional teaching and learning to a PBL-based approach. Based on a literature review on PBL theories, methods and tools, and observations made in classrooms and group rooms at Aalborg University, this chapter discusses possible aspects to consider when changing traditional computer science classrooms. A case is then used to demonstrate the process of changing traditional teaching and learning of a computer science course named Visual Programming and Applications to a PBL-based approach.

DOI: 10.4018/978-1-5225-9961-6.ch003

INTRODUCTION

Software development is a fast-developing domain. It is developing new techniques, frameworks, and platforms every day, and expanding its territory into other industrial domains every year. Projects in such a territory can always require new knowledge and may be very complex involving lots of people from different areas. Real-life software development is calling for practical developers that can adapt to the expanding body of knowledge in software development, and have skills and competencies beyond technical ones such as problem-solving, collaboration, project management and critical thinking (Dolog, Thomsen, & Thomsen, 2016).

However, the teaching and learning in today's Computer Science classrooms are still somewhat traditional. Computer Science courses often consist of teacher-centered long lectures and written-based exams. These traditional learning and assessment methods may be effective at low-level learning outcomes in Bloom's taxonomy such as remembering but may have trouble at high-level learning outcomes such as evaluating and creating (Anderson & Krathwohl, 2001). Meanwhile, such non-practical ways of learning may also have trouble in fostering non-technical transferable skills such as collaboration and project management, as these skills are highly situated and require lots of practice and participation (Kolmos & de Graaff, 2014). Fostering practical skills of developers requires changes in the teaching and learning in today's Computer Science classrooms.

Problem-Based Learning (PBL) is a potential solution towards nurturing practical developers. PBL is a student-centered methodology in that learning takes place in the process of students solving problems. PBL is a complex training and closely related to a large bunch of pedagogical concepts (Kolmos & de Graaff, 2014). Typical learning principles involved in PBL include problem organized learning, interdisciplinary learning, team-based learning, self-directed learning, etc. Thus, PBL learning principles highly coincide with the industry's requirements on practical developers. Meanwhile, PBL provides a concrete set of methodologies, methods, and tools to realize the social requirements.

This chapter discusses the challenges posed by real-life software development on Computer Science education of modern universities, and how to face these challenges by changing traditional teaching and learning to a PBL based approach. This chapter reviewed the literature on PBL and related learning theories and observations made in Aalborg University classrooms and group rooms to discuss possible ways to change the traditional teaching and learning of Computer Science courses. This chapter also introduces a case of reforming a traditional Computer Science course named Visual Programming and Applications to demonstrate a possible process of changing traditional Computer Science courses.

31 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/fostering-practical-developers-in-computer-science-classrooms/229373

Related Content

Putting Course Management Systems Behind Us

Robby Robson (2005). *Course Management Systems for Learning: Beyond Accidental Pedagogy* (pp. 282-295).

www.irma-international.org/chapter/putting-course-management-systems-behind/7187

Key Success Factors in an Online English Learning Environment From Tertiary Learners' Perspectives

Atima Kaewsa-ard (2022). *International Journal of Online Pedagogy and Course Design* (pp. 1-15).

www.irma-international.org/article/key-success-factors-in-an-online-english-learning-environment-from-tertiary-learners-perspectives/311436

Communities of Practice

Susan Gebhard (2008). *Encyclopedia of Information Technology Curriculum Integration* (pp. 124-130).

www.irma-international.org/chapter/communities-practice/16691

A Virtual-Reality Approach for the Assessment and Rehabilitation of Multitasking Deficits

Otmar Bock, Uwe Drescher, Wim van Winsum, Thomas F. Kesnerusand Claudia Voelcker-Rehage (2020). *Learning and Performance Assessment: Concepts, Methodologies, Tools, and Applications* (pp. 800-811).

www.irma-international.org/chapter/a-virtual-reality-approach-for-the-assessment-and-rehabilitation-of-multitasking-deficits/237556

Correlations Between the UN SDGs and Educational Technology From the Perspective of Taiwan's Educational Innovation

Shin-Jia Ho (2022). *International Journal of Online Pedagogy and Course Design* (pp. 1-7).

www.irma-international.org/article/correlations-between-the-un-sdgs-and-educational-technology-from-the-perspective-of-taiwans-educational-innovation/304081