

Chapter 10

Applying Online Learning in Software Engineering Education

Zuhoor Abdullah Salim Al-Khanjari
Sultan Qaboos University, Oman

ABSTRACT

Software Engineering education involves two learning aspects: (1) teaching theoretical material and (2) conducting the practical labs. Currently, Software Engineering education faces a challenge, which comes from the new learning opportunities afforded by the Web technologies. Delivering a Software Engineering curriculum by online distance learning requires innovative and flexible approaches to present and manage the theoretical and practical learning materials. E-Learning could support Software Engineering education through utilizing special e-Learning concepts, techniques, and tools. E-Learning could also change the mode of teaching from knowledge-as-transmission to knowledge-as-construction. This is called “Software Engineering e-Learning.” This chapter provides a review on Software Engineering education and e-Learning technology. It explores the need to adopt a Software Engineering e-Learning model to help the facilitators/instructors prepare and manage the online Software Engineering courses. This chapter also addresses how e-Learning environment could simplify the application of the constructivist learning model towards Software Engineering education.

1. INTRODUCTION

Software engineering as a computer science discipline can be deliberated from two perspectives: (1) “Software Engineering e-Learning” and (2) “e-Learning Software Engineering.” The former perspective arises when e-Learning concepts, techniques, and tools are used to support software engineering education. However, the later perspective takes place when the software engineering development methodologies and techniques are used for the development of e-Learning Management Systems (LMS). Both aspects have been thoroughly investigated in a large joint project, termed MuSoft (Multimedia in Software Engineering) (Doberkat et al., 2005).

DOI: 10.4018/978-1-5225-3923-0.ch010

Software products are significant assets in daily life of all organizations. Therefore, providing high quality software represents the goal of any software development and maintenance activity. To achieve this objective, it is important to educate IT professionals with high standards of development techniques.

Software Engineering is the sub-discipline and the umbrella of computer science that incorporates various accepted methodologies to design software. It deals with concepts, techniques and tools for supporting the development of high quality software systems. Teaching Software Engineering courses involves the consideration of theoretical and practical parts. In traditional ways, instructors use lectures to discuss the theoretical aspects and use labs to show students how to use and apply the concepts and techniques practically. We consider this situation in the lower level Software Engineering courses. However, the concern here is teaching the high level Software Engineering courses, which involve putting hands on practical work. Although, this is seen as easy as the lower level courses, the fact is there is a need to develop good and professional Software Engineering development team members. Therefore, teaching Software Engineering is considered a difficult process. The reason for this is that the concepts of Software Engineering cannot be realized and understood without a good practice and training on real life problems. The instructors are facing challenges to present this component in normal classrooms because of the need for students to work on industrial projects.

E-Learning concepts and techniques represent a good compensation solution to the above mentioned problem. E-Learning technology empowers the educational institutions to achieve better learning outcomes in a cost-efficient way. Multimedia techniques in particular support teaching of complex data structures and algorithms. Also, animations demonstrate the usage of complex software development tools in an interactive and simple manner. Students can benefit from the e-Learning environment in that the demonstration is available online and can be repeated as needed.

This chapter explores the use of e-Learning technologies to support the delivery of Software Engineering courses online. New technological advances such as high-speed Internet connections, virtual classrooms and virtual labs have removed the barriers limiting the learning opportunities of distance education. This chapter addresses some of the important issues related to utilizing e-Learning in Software Engineering education. It provides a background on the topic. It describes strategic learning models, e-Learning and Learning Management Systems. Further it discusses the importance of employing Software Engineering e-Learning model in Software Engineering education. Finally, it provides concluding remarks and future suggestions.

2. BACKGROUND

Over the last decade, the nature of education has changed significantly following the advances of information technology. The importance of utilizing computers and technology in education was predicted by many researchers (Daniel 1996; Crossman, 1997), who have advocated enhancing student learning by using digital tools, i.e. the e-Learning. The biggest advantage of e-Learning is that it gives students active learning opportunities. For this purpose, the researchers and academics have used several terminologies in education. These technologies include Computer Assisted Instruction (CAI), Computer Based Training (CBT) and Computer Assisted Learning (CAL) (Grieve, 1992; McDonough et al., 1994; Serdiukov, 2000).

Considerable research has been done to study e-Learning. For example, Tsai and Tsai studied the importance of e-Learning (Tsai & Tsai, 2003). Barra & Usman (2013) stressed on the importance of

13 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/applying-online-learning-in-software-engineering-education/192880

Related Content

Exceptions in Ontologies: A Theoretical Model for Deducing Properties from Topological Axioms

Christophe Jouis, Julien Bourdaillet, Bassel Habiband Jean-Gabriel Ganascia (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 61-81).

www.irma-international.org/chapter/exceptions-ontologies-theoretical-model-deducing/62435

Towards a Holistic Approach to Fault Management : Wheels Within a Wheel

Moises Goldszmidt, Miroslaw Malek, Simin Nadjm-Tehrani, Priya Narasimhan, Felix Salfner, Paul A. S. Wardand John Wilkes (2012). *Dependability and Computer Engineering: Concepts for Software-Intensive Systems* (pp. 1-10).

www.irma-international.org/chapter/towards-holistic-approach-fault-management/55321

Developing Secure Software Using UML Patterns

Holger Schmidt, Denis Hateburand Maritta Heisel (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 741-781).

www.irma-international.org/chapter/developing-secure-software-using-uml-patterns/192900

Flexible Orchestration of Tools in E-Collaboration: Case Studies Analyzing the Developer, the Teacher, and the Student Perspectives

Ioannis Magnisalisand Stavros Demetriadis (2019). *Handbook of Research on Technology Integration in the Global World* (pp. 208-232).

www.irma-international.org/chapter/flexible-orchestration-of-tools-in-e-collaboration/208800

Selection of Representative Feature Training Sets With Self-Organized Maps for Optimized Time Series Modeling and Prediction: Application to Forecasting Daily Drought Conditions With ARIMA and Neural Network Models

Elizabeth McCarthy, Ravinesh C. Deo, Yan Liand Tek Maraseni (2018). *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering* (pp. 446-464).

www.irma-international.org/chapter/selection-of-representative-feature-training-sets-with-self-organized-maps-for-optimized-time-series-modeling-and-prediction/206761