

Chapter 99

The Challenges of Teaching and Learning Software Programming to Novice Students

Seyed Reza Shahamiri

Manukau Institute of Technology, New Zealand

ABSTRACT

As software applications play a vital role in our daily lives, there is a need to have more skilled programmers to create such software products. There are various emerging disciplines like internet of things, driver-less cars, data science, software-defined networks, etc. that demand more programmers in the near future; hence, a more promising carrier for software developers is expected. Nevertheless, we have seen a low success rate in programming classes where some students lose interest in learning the required skills as they find programming a software a very challenging task; it has been reported that the fail rate of first programming papers in university computer science programs can be up to 60%. This chapter looks at some of the issues in regards to teaching and learning software programming and the nature of programming that may negatively influence the students' attention. Remedies to tackle the issues are also provided emphasizing on blended delivery using the technologies to facilitate the learning.

INTRODUCTION

Software Engineering is an engineering discipline that involves with all aspect of software development that applies engineering approaches in order to deliver high quality software products (Pressman, 2005). One of the important stages of software production is coding in which software blueprints are realized via a programming language; a programming language is a language that is understandable for computers (Sommerville, 2007). Table 1 shows the general phases of software development. Coding and programming language skills are required from phase four but knowledge and understanding of programming language is very helpful in previous phases in order to successfully complete a software project.

There has been a dramatic demand increase for software applications that promises a rewarding carrier for those who poses the required skills. With the fast advances in technology and emerging ones like

DOI: 10.4018/978-1-5225-7598-6.ch099

The Challenges of Teaching and Learning Software Programming to Novice Students

Table 1. Generic software development life cycle phases

No	Name	Description
1	<i>Planning</i>	The planning phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it.
2	<i>Analysis</i>	The analysis phase answers the questions of who will use the system, what the system will do, and where and when it will be used.
3	<i>Design</i>	Based on the user requirements, planning and the detailed analysis, the new system must be designed i.e. a blueprint of the system is created by designing the technical architecture.
4	<i>Implementation</i>	Actually implementing the designed system; writing software programs using software languages.
5	<i>Testing</i>	Checking whether the implemented software works according to specified requirements; fixing bugs/errors.
6	<i>Maintenance</i>	To ensure that the implemented system is properly functioning as per the requirements.

driverless cars, Internet of Things (IoT) (Pandya & Champaneria, 2015), Big Data (Sharma & Mangat, 2015), Software Defined Networks (Bizanis & Kuipers, 2016) etc. it is expected that more software programmers will be required in the future. As the result of such demands, various online and offline courses to introductory programming have been provided.

While knowing software coding is a very useful skill, it is difficult to learn programming especially at the beginning level since acquisition of complex new knowledge, associated strategies, and practical skills are required (Robins, Rountree, & Rountree, 2003). Software development courses are generally among difficult subjects and have low pass rates; according to (Dehnadi & Bornat, 2006) the fail rate of first programming papers in university computer science programmes can be up to 60 percent.

However, what are the reasons that make learning and teaching programming difficult? Why do students find it so challenging? And, why the success rates of programming classes are amongst the lowest in computer science papers? The next section tries to identify the reasons and issues that make learning introductory programming challenging. Next, some teaching and learning guidelines are provided to facilitate some of the identified challenges. Finally, recommendations for future studies are provided. The guidelines provided in this chapter are based on the literature and the author's extensive experience in teaching software programming.

BACKGROUND

This section provides a background of the problem and explains the issues in software programming teaching and learning.

Since software is intangible and cannot be seen or touched, creating software products can be a very complex task in comparison to most of the other engineering products; a software project can fail easily and lead to poor quality and unreliable products (Ammann & Offutt, 2008). On the other hand, software applications play very important and critical roles in modern life because they control vital operations that require attributes such as security, reliability, performance, etc., qualities that are hard to achieve (Spillner, Linz, & Schaefer, 2007).

Because of the intangible nature of software, and since students cannot directly sense what they have created, it can become very complex to them to successfully implement, debug and verify the product. In addition, it is more challenging to teach and learn the introduction of coding since this is where the

6 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/the-challenges-of-teaching-and-learning-software-programming-to-novice-students/214705

Related Content

Online Authentication Using Smart Card Technology in Mobile Phone Infrastructure

Teddy Mantoro, Admir Milišić and Media Ayu (2011). *International Journal of Mobile Computing and Multimedia Communications* (pp. 67-83).

www.irma-international.org/article/online-authentication-using-smart-card/58906

Mobile E-Commerce as a Strategic Imperative for the New Economy

M. Raisinghani (2007). *Encyclopedia of Mobile Computing and Commerce* (pp. 472-480).

www.irma-international.org/chapter/mobile-commerce-strategic-imperative-new/17120

Semantic Web Services and Mobile Agents Integration for Efficient Mobile Services

Vasileios Baousis, Vassilis Spiliopoulos and Elias Zavitsanos (2009). *Mobile Computing: Concepts, Methodologies, Tools, and Applications* (pp. 2936-2956).

www.irma-international.org/chapter/semantic-web-services-mobile-agents/26703

Mobile Health Systems for Bipolar Disorder: The Relevance of Non-Functional Requirements in MONARCA Project

Oscar Mayora, Mads Frost, Bert Arnrich, Franz Gravenhorst, Agnes Grunerbl, Amir Muaremi, Venet Osmani, Alessandro Puiatti, Nina Reichwaldt, Corinna Scharnweber and Gerhard Troster (2014). *International Journal of Handheld Computing Research* (pp. 1-12).

www.irma-international.org/article/mobile-health-systems-for-bipolar-disorder/111344

Perceived Mobile Information Security and Adoption of Mobile Payment Services in China

Fei Gao, Pei-Luen Patrick Rau and Yubo Zhang (2018). *Mobile Commerce: Concepts, Methodologies, Tools, and Applications* (pp. 1179-1198).

www.irma-international.org/chapter/perceived-mobile-information-security-and-adoption-of-mobile-payment-services-in-china/183334