The Case for Non-linguistic Approach to Teaching Engineering Thinking

Leon Brânzan[1]

[1]Dept. of Informatics and System Engineering, Faculty of Computers, Informatics and Microelectronics, Technical University of Moldova, str. Studenţilor, 9/7, b. 3, MD-2045, Chișinau, Republic of Moldova,
leon.brinzan@iis.utm.md, https://fcim.utm.md/

ABSTRACT

Software engineering is a multifaceted discipline. Mastering it requires memorizing a lot of factual information as well as learning a particular set of skills that enable the learner to tackle complex engineering tasks in a more efficient manner. It also requires the knowledge of at least one programming language. This makes teaching software engineering efficiently difficult. The traditional way of teaching programming (and by extension - engineering) to students heavily relies on delivering information in the form of lectures and via textbooks. This paper argues that lectures are becoming less and less effective when it comes to teaching programming, and that there are methods of doing it using software solutions which take into account how the human brain processes code. This paper will provide several concrete examples of how software enables students to learn the "spirit of engineering and problem solving". Lastly, it argues that teaching programming without using textual explanations can be more effective in some contexts, and identifies several key aspects of using software that allow teaching major programming concepts without using textual information in order to make the whole learning process more efficient.