

# Can we make Software Engineering Education better by applying Learning Theories?

Sridhar Chimalakonda, Kesav V. Nori

*International Institute of Information Technology – Hyderabad, Hyderabad*

*sridhar\_ch@research.iiit.ac.in, kesav.nori@gmail.com*

*Today, software is being developed and maintained not just by engineers educated formally but also by people from widely different backgrounds including children, knowledge workers among many others. However, a number of studies indicate that most of these people are not equipped with enough software engineering skills leading to low-quality software. On the other hand, an analysis of literature reveals that the significant potential of learning theories (LTs) like Learning by Doing, Reflection, Model-based and so on has not been fully utilized by Software Engineering Education (SEE) and Training community. In this context, the main thrust of this poster is to emphasize the need for using vast amount of LTs (nearly 110) available to effectively and efficiently teach software engineering and help in producing quality software engineers. In this poster, we motivate the need for use of LTs in SEE, while summarizing significant ones and their use in SE. The poster also presents a mapping of key SE topics and their proposed LTs while emphasizing the role of context in using LTs. The poster further presents teaching Software Product Lines as an illustrative example. Finally, future work of this poster includes designing modeling languages to represent LTs and mapping them to topics in SWEBOOK and GSwERC (software engineering curriculum) contributing to SEE community and Software Engineering at large.*

The inevitable reliance on software in current society and the need for software development and maintenance at a massive scale have forced the industry to employ people from widely different backgrounds and skills during entire life cycle of software development. Emerging disciplines like end-user-software-engineering encourage end-users to develop and author software for their needs. This presents the major challenge of equipping these people with software engineering skills such that they deliver quality software. The responsibility of creating software engineers from undergrads, graduates and professionals involved in software development lies with the SEE community. We analyzed the existing body of SEE and found that a number of approaches are employed to create software engineers. Notable approaches include practice-oriented approaches focusing on projects that help students to understand the challenges in industry projects, approaches focusing on theory wherein specific topics that are of importance to practice are taught and game-based approaches focusing on simulating specific software engineering topics like Requirements Engineering, Personal Software Process are common place. *During our analysis, we have also observed that most of the educators and instructors used learning theories implicitly hence leaving the process of teaching to the expertise of the instructor.* Another important direction is the use of eLearning and collaboration as a medium of instruction to teach and learn software engineering skills from across the globe. Learning theories such as Learning by Doing, Discovery Learning and Learning through Reflection are more relevant in the context of SEE. Applying learning theories to teach software engineering has been successfully discussed in [1] however there are only a few cases where this has been done. Our poster is another case of strengthening this need of using LTs in SEE and extends the mapping of LTs to SE topics presented in [1] by emphasizing on context.

[1] Emily Oh Navarro and André van der Hoek, On the Role of Learning Theories in Furthering Software Engineering Education, *In H.J.C. Ellis, S.A. Demurjian, and J.F. Naveda (Eds), Software Engineering: Effective Teaching and Learning Approaches and Practices, IGI Global, 2008.*