Corpus of Literature for a Systematic Mapping Study on Practical Approaches for Teaching Software Engineering

Maíra R. Marques  
Computer Science Department  
Universidad de Chile  
Santiago, Chile  
mmarques@dcc.uchile.cl

Alcides Quispe  
Computer Science Department  
Universidad de Chile  
Santiago, Chile  
aquispe@dcc.uchile.cl

Sergio F. Ochoa  
Computer Science Department  
Universidad de Chile  
Santiago, Chile  
sochoa@dcc.uchile.cl

SUMMARY

There is no consensus on universities around the world on how to teach software engineering. Each university uses their own methods and approaches based on the experiences of their instructors. It is worth knowing where each one own university stands and what are other universities doing about the subject. This technical report presents the corpus of literature used in a systematic mapping study on practical approaches for teaching software engineering. Such a study was conducted following the process proposed by Petersen et al. [1] (Figure 1).

Figure 1 - Systematic Mapping Process by Petersen et al. [1]

Systematic mapping studies are intended to give an overview of a specific topic; in our case, it is the practical approaches for teaching software engineering. Typically, a systematic mapping study is classified as a secondary study and indicates the quality and quantity of work already done by the research community in the area. The papers used in the mapping study are selected and classified according to a set of the research questions defined by the authors. The research questions to be answered in this systematic mapping study are the following:

**RQ1**: What are the main approaches used to address the practical experiences in software engineering education?

**RQ2**: Is there an emerging tendency to address this challenge?

**RQ3**: Which software process models are used to support the practical experiences in software engineering courses?

**RQ4**: Have the universities changed the way of conducting these experiences over the years?

**RQ5**: What are the main forums to seek information on practical approaches for teaching software engineering?

One of the main goals of a systematic mapping study is to categorize an area of knowledge providing evidence that there is information and possible clusters where further research could be done.

The sources of literature used on this study were the following:

- ACM Digital Library (http://dl.acm.org)
- IEEExplore (http://ieeexplore.ieee.org/Xplore/home.jsp)
- Web of Knowledge (http://www.webofknowledge.com/)
- SpringerLink (http://www.springer.com)
- Science Direct (http://www.sciencedirect.com)
- Wiley International (http://onlinelibrary.wiley.com)

As a result of this study 173 papers were selected and classified; next section presents the list of papers involved in this study. The information retrieval process was done from 1st to 30rd October 2013. The main results of this study were the following:

- The majority of the reports are worried that students have a “practical experience”, but authors do not mention clearly any pedagogical approach (such as PBL – problem based learning).
- There are studies reporting the use of game learning, cases studies, simulation, inverted classroom, maintenance, service learning, open source projects as teaching approaches.
- The use of traditional approaches is reported in the past, but there are only one report made more recently.
- The use of a process development is not a clear concern stated in the studies, just a small part of the studies report its use (69 from 173) and the Agile is the most used among them.
Conferences are the most used forum to publish studies in the area (72%). And one third of the studies have been published in the last five years.

**PRIMARY STUDIES ON PRACTICAL APPROACHES FOR TEACHING SOFTWARE ENGINEERING**


34. J. Vanhanen, T.O.A Lehtinen, and C. Lassenius, “Teaching real-world software engineering through a capstone project course with industrial customers,” Proceedings of the
64. S. Ludi, “The benefits and challenges of using educational game projects in an undergraduate software engineering course,” presented at the Workshop on Games and Software Engineering, 2011.


REFERENCES