

# 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).

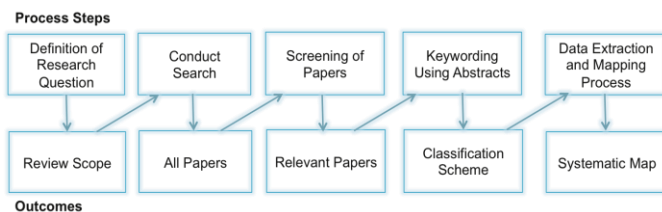


Figure 1 - Systematic Mapping Process by Petersen et al. [1]

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>)

IEEE Xplore (<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

1. M. B. Blake, "A student-enacted simulation approach to software engineering education," *IEEE Transactions of Education*, vol. 46, no. 1, pp. 124–132, Feb. 2003.
2. M. Bernreuther and H.-J. Bungartz, "Learning by doing: Software projects in CSE education," *Practical Aspects of Knowledge Management*, 2006, vol. 3992, pp. 161–168.
3. D. F. Rico and H. H. Sayani, "Use of Agile Methods in Software Engineering Education," presented at the Agile Conference, 2009, pp. 174–179.
4. E. D. Ragan, S. Frezza, and J. Cannell, "Product-based learning in software engineering education," presented at the *Frontiers in Education*, 2009, pp. 1–6.
5. L. H. Werth, "Software process improvement for student projects," presented at the *Frontiers in Education*, 1995, vol. 1, pp. 2–2b1.4.
6. H. B. Christensen, "A story-telling approach for a software engineering course design," presented at the *ACM SIGCSE Bulletin*, 2009.
7. C. Bunse, R. L. Feldmann, and J. Dörr, "Agile methods in software engineering education," *Agile Processes in Software Engineering*, 2004.
8. Y. A. D. H. A. H. Y. Wang, "A case study of Problem-Based Learning instruction design under web-based environment," presented at the *International Conference on Education Technology and Computer*, 2010, vol. 1, pp. V1–586–V1–591.
9. T. P. Way, "A company-based framework for a software engineering course," *ACM SIGCSE Bulletin*, 2005.
10. B. Meyer and M. Piccioni, "The Allure and Risks of a Deployable Software Engineering Project: Experiences with Both Local and Distributed Development," presented at the *Conference on Software Engineering Education and Training*, 2008, pp. 3–16.
11. Rusu, A. Rusu, R. Docimo, and C. Santiago, "Academia-academia-industry collaborations on software engineering projects using local-remote teams," presented at the *SIGCSE Technical Symposium on Computer Science Education*, 2009.
12. F. Ciccozzi and I. Crnkovic x0301, "Performing a Project in a Distributed Software Development Course: Lessons Learned," presented at the *Global Engineering Education Conference*, 2010, pp. 187–191.
13. J. S. Collofello and S. N. Woodfield, "A project-unified software engineering course sequence," *ACM SIGCSE Bulletin*, 1982.
14. H. Ziv and S. Patil, "Capstone Project: From Software Engineering to 'Informatics'," presented at the *Conference on Software Engineering Education and Training*, 2010.
15. Rusu and M. Swenson, "An industry-academia team-teaching case study for software engineering capstone courses," presented at the *Frontiers in Education*, 2008.
16. J. E. Tomayko, "Carnegie Mellon's software development studio: a five year retrospective," presented at the *Conference on Software Engineering Education*, 1996, pp. 119–129.
17. K. Garg and V. Varma, "A Study of the Effectiveness of Case Study Approach in Software Engineering Education," presented at the *Conference on Software Engineering Education and Training*, 2007, pp. 309–316.
18. Stringfellow and D. Mule, "Smartphone applications as software engineering projects," *Journal of Computing Sciences in Colleges*, 2013.
19. R. Razali and M. Chitsaz, "Cases development for teaching software engineering," presented at the *International Conference on Education Technology and Computer*, 2010, vol. 2.
20. R. McCartney, S. S. Gokhale, and T. M. Smith, "Evaluating an early software engineering course with projects and tools from open source software," presented at the *International Computing Education Research*, 2012.
21. W. Liew, "Teaching software development skills early in the Curriculum through software engineering," *ACM SIGCSE Bulletin*, 2005.
22. D.C. Peixoto, R.M. Possa, R.F. Resende, and C.I. Padua, "Challenges and issues in the development of a Software Engineering simulation game," presented at the *Frontiers in Education*, 2012, pp. 1–6.
23. V. Varma and K. Garg, "Case studies: the potential teaching instruments for software engineering education," presented at the *International Conference on Quality Software*, 2005, pp. 279–284.
24. W. Gurney, P. Maas, and G. May, "An experiment in software engineering education," *Software Engineering Journal*, vol. 2, no. 4, pp. 127–132, 1987.
25. R. L. A. de A. E. S. Neto, "Five years of lessons learned from the Software Engineering course: Adapting best practices for Distributed Software Development," *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD)*, 2012, pp. 6–10, 2012.
26. J. I. Maletic, A. Howald, and A. Marcus, "Incorporating PSP into a traditional software engineering course: an experience report," presented at the *Conference on Software Engineering Education and Training*, 2001, pp. 89–97.
27. P. Desai and G. H. Joshi, "Activity based teaching learning in software engineering - An experience," presented at the *Engineering Education: Innovative Practices and Future Trends (AICERA)*, 2012 *IEEE International Conference on*, 2012, pp. 1–6.
28. T. Bassler, M. Deininger, K. Schneider, and J. Schwill, "SESAM-simulating software projects," presented at the *Software Engineering and Knowledge Engineering, International Conference on*, 1992, pp. 608–615.
29. C. Keen, C. Lockwood, and J. Lamp, "A client-focused, team-of-teams approach to software development projects," presented at the *International Conference Software Engineering: Education & Practice*, 1998, pp. 34–41.
30. Drappa and J. Ludewig, "Simulation in software engineering training," presented at the *International Conference on Software Engineering*, 2000.
31. H. Mayr, "Teaching software engineering by means of a 'virtual enterprise'," presented at the *International Conference on Computer Science & Education*, 1997.
32. Shukla and L. Williams, "Adapting extreme programming for a core software engineering course," presented at the *Conference on Software Engineering Education and Training*, 2002, pp. 184–191.
33. M. J. Oudshoorn and K. J. Maciunas, "Experience with a project-based approach to teaching software engineering," presented at the *Software Education Conference*, 1994, pp. 220–225.
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*

- International Workshop on Software Engineering Education based on Real-World Experiences, 2012, pp. 29–32.
35. K. Cox and K. T. Phalp, “Practical experience of eliciting classes from use case descriptions,” *Journal of Systems and Software*, vol. 80, no. 8, pp. 1286–1304, 2007.
  36. J. Pittges, J. D. Chase, and T. Lewis, “Supporting Live Student Projects: Lessons Learned from the Small Project Support Center at Radford University,” presented at the International Conference on Information Technology: New Generations, 2009.
  37. G. W. Morgan and F. A. Lear, “The role of a software engineering project within an undergraduate applied computing degree,” presented at the Software Education Conference, 1994, pp. 230–236.
  38. J. Chao, “Balancing hands-on and research activities: a graduate level agile software development course,” presented at the Agile Conference, 2005, pp. 306–311.
  39. M. Gnatz, L. Kof, F. Prilmeier, and T. Seifert, “A practical approach of teaching Software Engineering,” presented at the Conference on Software Engineering Education and Training, 2003, pp. 120–128.
  40. K. R. Pierce, “Teaching software engineering principles using maintenance-based projects,” presented at the Conference on Software Engineering Education and Training, 1997, pp. 53–60.
  41. M. J. Sebern, “The software development laboratory: incorporating industrial practice in an academic environment,” presented at the Conference on Software Engineering Education and Training, 2002, pp. 118–127.
  42. L. Al-Abdulkarim and M. Al-Yahya, “Teaching Software Engineering Principles Using Micro Project with Eclipse (MPE) Activity: An Evaluation,” presented at the Software Engineering Advances, 2009.
  43. Broman, K. Sandahl, and M. A. Baker, “The company approach to software engineering project courses,” *IEEE Transactions on Education*, 2011.
  44. P. Brazier, A. Garcia, and A. Vaca, “A software engineering senior design project inherited from a partially implemented software engineering class project,” presented at the Frontiers in Education, 2007.
  45. Y. Ouyang and S. S. Wang, “Real-world object-oriented design experience for computer science students,” presented at the Frontiers in Education, 1999, vol. 2, p. 13.
  46. T. Baar, “Improving software engineering education by modeling real-world implementations,” presented at the Educators’ Symposium, 2012.
  47. C. L. Bullard, I. Caldwell, J. Harrell, and C. Hinkle, “Anatomy of a software engineering project,” *ACM SIGCSE Bulletin*, 1988.
  48. H. J. C. Ellis, “An experience in collaborative learning: observations of a software engineering course,” presented at the Frontiers in Education, 2000, vol. 1, pp. –T2C/6 vol.1.
  49. M. Wainer, “Adaptations for teaching software development with extreme programming: An experience report,” *Extreme Programming and Agile Methods-XP/Agile ...*, 2003.
  50. J. C. Schlimmer, J. B. Fletcher, and L. A. Hermens, “Team-oriented software practicum,” *IEEE Transactions on Education*, vol. 37, no. 2, pp. 212–220, May 1994.
  51. B. Turhan and A. Bener, “A Template for Real World Team Projects for Highly Populated Software Engineering Classes,” presented at the International Conference on Software Engineering, 2007, pp. 748–753.
  52. J. D. Bayliss and S. Strout, “Games as a ‘flavor’ of CS1,” presented at the SIGCSE Technical Symposium on Computer Science Education, New York, New York, USA, 2006, vol. 38, no. 1, pp. 500–504.
  53. O. Gotel, V. Kulkarni, L. C. Neak, C. Scharff, and S. Seng, “Introducing global supply chains into software engineering education,” presented at the International Conference on Software engineering Approaches for Offshore and Outsourced Development, 2007.
  54. P. A. Laplante, “An agile, graduate, software studio course,” *IEEE Transactions on Education*, vol. 49, pp. 417–419, 2006.
  55. T. M. Connolly, M. Stansfield, and T. Hainey, “An application of games-based learning within software engineering,” *British Journal of Educational Technology*, vol. 38, no. 3, pp. 416–428, 2007.
  56. R. E. Seviora, “A real-time project for software engineering course,” presented at the Software Engineering for Real Time Systems, International Conference on, 1989, pp. 65–69.
  57. Shakshuki, R. Giles, and O. Prabhu, “Experience in teaching an introductory software engineering course,” presented at the Information Technology Based Higher Education and Training, 2005. ITHET 2005. 6th International Conference on, 2005.
  58. D. Coppit and J. M. Haddox-Schatz, “Large team projects in software engineering courses,” *ACM SIGCSE Bulletin*, 2005.
  59. J. C. Ellis, “Software engineering at a distance,” presented at the Conference on Software Engineering Education, 1998, pp. 23–34.
  60. Ruhe, “Experience Factory-based professional education and training,” presented at the Conference on Software Engineering Education and Training, 1999, pp. 62–72.
  61. C. Knight and T. B. Horton, “Evaluating a software engineering project course model based on studio presentations,” presented at the Frontiers in Education, 2005.
  62. Schilling and R. Klamma, “The difficult bridge between university and industry: a case study in computer science teaching,” *Assessment & Evaluation in Higher ...*, 2010.
  63. M. Clifton, “An industry approach to the software engineering course,” *ACM SIGCSE Bulletin*, 1991.
  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.
  65. R. Bareiss and M. Griss, “A story-centered, learn-by-doing approach to software engineering education,” presented at the SIGCSE Technical Symposium on Computer Science Education, New York, New York, USA, 2008, p. 221.
  66. J. Reed, “A parser for a small Ada-like language—a student software engineering project,” presented at the SIGCSE Technical Symposium on Computer Science Education, New York, New York, USA, 1985, pp. 241–254.
  67. N. Petalidis, “Adopting Lakatos in a software engineering course,” *International Journal of engineering Education*, vol. 24, pp. 738–746, 2008.
  68. T. B. Hilburn, M. Towhidnejad, S. Nangia, and L. Shen, “A Case Study Project for Software Engineering Education,” presented at the Frontiers in Education, 2006, pp. 1–5.
  69. V. Mahnic, “A Capstone Course on Agile Software Development Using Scrum,” *IEEE Transactions on Education*, vol. 55, no. 1, pp. 99–106, 2012.
  70. D. M. Nguyen, T. V. Truong, and N. B. Le, “Deployment of Capstone Projects in Software Engineering Education at Duy Tan University as Part of a University-Wide Project-Based Learning Effort,” presented at the Learning and Teaching in Computing and Engineering, 2013, pp. 184–191.
  71. P. N. Robillard, P. Kruchten, and P. d’Astous, “YOOPEEDOO (UPEDU): A Process for Teaching Software Process,”

- presented at the Conference on Software Engineering Education and Training, 2001, pp. 18–26.
72. S. Henry, "A project oriented course on software engineering," *ACM SIGCSE Bulletin*, 1983.
  73. Richardson and Y. Delaney, "Problem Based Learning in the Software Engineering Classroom," presented at the Conference on Software Engineering Education and Training, 2009, pp. 174–181.
  74. D. J. Bagert, J. M. Gregory, S. A. Mengel, and L. R. Heinze, "Engineering education innovation with software engineering projects," presented at the *Frontiers in Education*, 2002, vol. 3, pp. S3G–13–S3G–16 vol.3.
  75. Emam and M. G. Mostafa, "Using game level design as an applied method for Software Engineering education," presented at the *International Conference on Computer Games*, 2012, pp. 248–252.
  76. Coupal and K. Boechler, "Introducing agile into a software development Capstone project," presented at the *Agile Conference*, 2005, pp. 289–297.
  77. E. S. Monsalve, V. M. B. Werneck, and J. C. Sampaio do Prado Leite, "Teaching software engineering with SimULES-W," presented at the Conference on Software Engineering Education and Training, 2011, pp. 31–40.
  78. P. Flener, "Realism in project-based software engineering courses: rewards, risks, and recommendations," presented at the *International Conference on Computer and Information Sciences*, 2006.
  79. C. Wohlin, "Meeting the challenge of large-scale software development in an educational environment," *SEDC*, pp. 40–52, 1997.
  80. R. G. Epstein, "A Software Engineering Course with an Emphasis on Software Processes and Security," presented at the Conference on Software Engineering Education and Training, 2008, pp. 67–73.
  81. Richardson, L. Reid, S. B. Seidman, B. Pattinson, and Y. Delaney, "Educating software engineers of the future: Software quality research through problem-based learning," presented at the Conference on Software Engineering Education and Training, 2011, pp. 91–100.
  82. F. Meawad, "The virtual agile enterprise: Making the most of a software engineering course," presented at the Conference on Software Engineering Education and Training, 2011, pp. 324–332.
  83. H. C. Nelson, T. Nute, and D. J. Rodjak, "Applying the spiral model: A case study in small project management," *Software Process: Improvement and Practice*, vol. 2, no. 4, pp. 239–251, 1996.
  84. Y.-P. Cheng and J. M.-C. Lin, "A Constrained and Guided Approach for Managing Software Engineering Course Projects," *IEEE Transactions on Education*, vol. 53, no. 3, pp. 430–436, Aug. 2010.
  85. R. B. Vaughn Jr, "A report on industrial transfer of software engineering to the classroom environment," presented at the Conference on Software Engineering Education and Training, 2000, pp. 15–22.
  86. J. Buchta, M. Petrenko, D. Poshyvanyk, and V. Rajlich, "Teaching Evolution of Open-Source Projects in Software Engineering Courses," presented at the *International Conference on Software Maintenance*, 2006, pp. 136–144.
  87. S. C. dos Santos, M. da Conceicao Moraes Batista, A. P. C. Cavalcanti, J. O. Albuquerque, and S. R. L. Meira, "Applying PBL in Software Engineering Education," presented at the Conference on Software Engineering Education and Training, 2009, pp. 182–189.
  88. B. R. V. Konsky and M. Robey, "A Case Study: GQM and TSP in a Software Engineering Capstone Project," presented at the Conference on Software Engineering Education and Training, 2005, pp. 215–222.
  89. J. Liu, J. Marsaglia, and D. Olson, "Teaching software engineering to make students ready for the real world," *Journal of Computing Sciences in Colleges*, 2002.
  90. T. Stevens, "Experiences teaching software engineering for the first time," presented at the Conference on Innovation and Technology in Computer Science Education, New York, New York, USA, 2001, pp. 77–80.
  91. V. Mohrenschildt and D. K. Peters, "The draw-bot: a project for teaching software engineering," presented at the *Frontiers in Education*, 1998, vol. 3, pp. 1022–1027.
  92. T. Smith, K. Cooper, and C. S. Longstreet, "Software engineering senior design course: experiences with agile game development in a capstone project," presented at the *Workshop on Games and Software Engineering*, 2011.
  93. J. Herold, T. D. Lynch, and R. Ramnath, "Student and Instructor Experiences in the Inverted Classroom," presented at the *Frontiers in Education*, 2012.
  94. J. Guo, "Group projects in software engineering education," *Journal of Computing Sciences in Colleges*, 2009.
  95. Brazier, "Process and product in a software engineering course: simulating the real world," presented at the *Frontiers in Education*, 1998, vol. 3, pp. 1292–1297.
  96. S. M. Weerawarana and S. Weerawardhana, "Leveraging a constructively aligned OBTL approach for software engineering education," presented at the *Teaching, Assessment and Learning for Engineering*, 2012, pp. –W1B–21.
  97. K. Garg and V. Varma, "Case Studies as Assessment Tools in Software Engineering Classrooms," presented at the Conference on Software Engineering Education and Training, 2009, pp. 8–11.
  98. Baker, E. O. Navarro, and A. van der Hoek, "An experimental card game for teaching software engineering," presented at the Conference on Software Engineering Education and Training, 2003, pp. 216–223.
  99. Baumgartner, "Using case studies to design and deliver technology-centered computing education courses: an innovative approach from an undergraduate information systems program in singapore," presented at the Conference on Innovation and Technology in Computer Science Education, New York, New York, USA, 2013, p. 189.
  100. M. Qiu and L. Chen, "A Problem-Based Learning Approach to Teaching an Advanced Software Engineering Course," presented at the *Education Technology and Computer Science International Workshop on*, 2010, vol. 3, pp. 252–255.
  101. S. Zuppiroli, P. Ciancarini, and M. Gabbrielli, "A Role-Playing Game for a Software Engineering Lab: Developing a Product Line," presented at the Conference on Software Engineering Education and Training, 2012, pp. 13–22.
  102. G. W. Hislop, "Scaffolding Student Work in Capstone Design Courses," presented at the *Frontiers in Education*, 2006, pp. 1–4.
  103. K. Claypool and M. Claypool, "Teaching software engineering through game design," *ACM SIGCSE Bulletin*, 2005.
  104. M. Žagar, I. Bosnić, and M. Orlić, "Enhancing software engineering education: a creative approach," presented at the *International Workshop on Software Engineering in East and South Europe*, New York, New York, USA, 2008, p. 51.
  105. L. Jaccheri and T. Osterlie, "Open Source Software: A Source of Possibilities for Software Engineering Education and Empirical Software Engineering," presented at the *Workshop*

- on Emerging Trends in FLOSS Research and Development, 2007, pp. 5–5.
106. M. Nordio, R. Mitin, and B. Meyer, “Advanced hands-on training for distributed and outsourced software engineering,” presented at the International Conference on Software Engineering, New York, New York, USA, 2010, vol. 1, pp. 555–558.
  107. V. Pieterse, L. Thompson, and L. Marshall, “An intensive software engineering learning experience,” presented at the Computer Science Education Research Conference, 2012.
  108. D. Port and B. Boehm, “Using a model framework in developing and delivering a family of software engineering project courses,” presented at the Conference on Software Engineering Education and Training, 2001, pp. 44–55.
  109. Hazzan and Y. Dubinsky, “Teaching a software development methodology: the case of extreme programming,” presented at the Conference on Software Engineering Education and Training, 2003, pp. 176–184.
  110. N. F. LeJeune, “Teaching software engineering practices with Extreme Programming,” *Journal of Computing Sciences in Colleges*, vol. 21, no. 3, Feb. 2006.
  111. S. M. Hadfield and N. A. Jensen, “Crafting a software engineering capstone project course,” *Journal of Computing Sciences in Colleges*, vol. 23, no. 1, Oct. 2007.
  112. E. Sweedyk and R. M. Keller, “Fun and games: a new software engineering course,” presented at the Conference on Innovation and Technology in Computer Science Education, New York, New York, USA, 2005, p. 138.
  113. J. Paynter and E. Sharkey, “Adventure cycles: a software engineering approach,” presented at the Australasian Conference on Computing Education, 2004, vol. 30.
  114. Baker, E. O. Navarro, and A. Van Der Hoek, “An experimental card game for teaching software engineering processes,” *Journal of Systems and Software*, 2005.
  115. J. Bolinger, M. Herold, R. Ramnath, and J. Ramanathan, “Connecting reality with theory; An approach for creating integrative industry case studies in the software engineering curriculum,” presented at the *Frontiers in Education*, 2011, pp. T4G-1–T4G-6.
  116. M. I. Alfonso and A. Botia, “An iterative and agile process model for teaching software engineering,” presented at the Conference on Software Engineering Education and Training, 2005.
  117. B. J. Cornelius, M. Munro, and D. J. Robson, “An approach to software maintenance education,” *Software Engineering Journal*, vol. 4, no. 4, pp. 233–236, 1989.
  118. D. P. Groth and E. L. Robertson, “It’s all about process: project-oriented teaching of software engineering,” presented at the Conference on Software Engineering Education and Training, 2001, pp. 7–17.
  119. T. Reichlmayr, “The agile approach in an undergraduate software engineering course project,” presented at the *Frontiers in Education*, 2003, vol. 3.
  120. D. E. Strode, S. L. Huff, B. Hope, and S. Link, “Coordination in co-located agile software development projects,” *Journal of Systems and Software*, vol. 85, no. 6, pp. 1222–1238, 2012.
  121. G. Lapalme and J.-F. Lamy, “An experiment in the use of Ada in course in software engineering,” presented at the SIGCSE Technical Symposium on Computer Science Education, New York, New York, USA, 1986, pp. 124–126.
  122. L. Olsen, “A service learning project for a software engineering course,” *Journal of Computing Sciences in Colleges*, 2008.
  123. Parrish, D. Cordes, C. Lester, and D. Moore, “Active learning and process assessment: two experiments in an Ada-based software engineering course,” presented at the Conference on TRI-Ada, New York, New York, USA, 1996, pp. 157–161.
  124. Anisetty and P. Young, “Collaboration problems in conducting a group project in a software engineering course,” *Journal of Computing Sciences in Colleges*, 2011.
  125. J. Ryoo, F. Fonseca, and D. S. Janzen, “Teaching Object-Oriented Software Engineering through Problem-Based Learning in the Context of Game Design,” presented at the Conference on Software Engineering Education and Training, 2008.
  126. T. Wang and Q. Zhu, “A Software Engineering Education Game in a 3-D Online Virtual Environment,” presented at the Education Technology and Computer Science International Workshop on, 2009, vol. 2, pp. 708–710.
  127. G. S. Owen, “An Ada-based software engineering course,” *ACM SIGCSE Bulletin*, 1989.
  128. J. G. Flowers, “Improving the Capstone project experience: a case study in software engineering,” presented at the Southeast regional conference, New York, New York, USA, 2008, p. 237.
  129. M. Postema, J. Miller, and M. Dick, “Including practical software evolution in software engineering education,” presented at the Conference on Software Engineering Education and Training, 2001, pp. 127–135.
  130. M. Halling, W. Zuser, M. Kohle, and S. Biffel, “Teaching the unified process to undergraduate students,” presented at the Conference on Software Engineering Education and Training, 2002, pp. 148–159.
  131. J. B. Fenwick, “Adapting XP to an academic environment by phasing-in practices,” presented at the *Practical Aspects of Knowledge Management*, 2003, vol. 2753, pp. 162–171.
  132. Butler, “A client/server case study for software engineering students,” presented at the Conference on Software Engineering Education and Training, 1999, pp. 156–165.
  133. Hanks, “Becoming Agile using Service Learning in the Software Engineering Course,” presented at the Agile Conference, 2007, pp. 121–127.
  134. Garcia and C. L. Pacheco, “Using TSPi and PBL to support software engineering education in an upper-level undergraduate course,” *Computer Applications in Engineering Education*, 2012.
  135. M. Daniels, X. Faulkner, and I. Newman, “Open ended group projects, motivating students and preparing them for the ‘real world’,” presented at the Conference on Software Engineering Education and Training, 2002, pp. 115–126.
  136. Concepcion, “Using an object-oriented software life-cycle model in the software engineering course,” *ACM SIGCSE Bulletin*, 1998.
  137. J. Chao and M. Randles, “Agile Software Factory for Student Service Learning,” presented at the Conference on Software Engineering Education and Training, 2009, pp. 34–40.
  138. Scharlau, “Games for teaching software development,” presented at the Conference on Innovation and Technology in Computer Science Education, 2013.
  139. Sowell, C. Gill, R. D. Chamberlain, and C. Grimm, “The active-learning transformation: a case study in software development and systems software courses,” *Journal of Computing Sciences in Colleges*, 2010.
  140. Tu, S. Tankashala, S. Oh, B. Becker, B. Horton, A. Kallem, Z. Yang, L. Hartman, and D. Wagner, “Turning real-world systems into verification-driven learning cases,” presented at the Conference on Software Engineering Education and Training, 2011, pp. 129–138.

141. J. Andersson and M. Edvinsson, "The Puppeteer - Directing Software Engineering Projects," presented at the International Conference on Computer Science & Education, 2008, vol. 5.
142. H. L. Lutfiyya and J. H. Andrews, "Experiences with a software maintenance project course," *IEEE Transactions on Education*, vol. 43, no. 4, pp. 383–388, n/a 2000.
143. L. M. Northrop, "Success with the project-intensive model for an undergraduate software engineering course," presented at the SIGCSE Technical Symposium on Computer Science Education, New York, New York, USA, 1989, pp. 151–155.
144. E. Veraart and S. L. Wright, "Software engineering education-adding process to projects: theory, practice and experience," presented at the Asia Pacific Software Engineering Conference, 1995, pp. 148–157.
145. J. H. Hayes, "Energizing software engineering education through real-world projects as experimental studies," presented at the International Conference on Computer Science & Education, 2002.
146. H. J. C. Ellis, R. A. Morelli, T. R. de Lanerolle, and G. W. Hislop, "Holistic Software Engineering Education Based on a Humanitarian Open Source Project," presented at the Conference on Software Engineering Education and Training, 2007, pp. 327–335.
147. P. Katz, "Software Engineering Practicum Course Experience," presented at the Conference on Software Engineering Education and Training, 2010, pp. 169–172.
148. Wortman, "Software Projects in an Academic Environment," *IEEE Trans. Software Eng.*, vol. 13, no. 11, pp. 1176–1181, 1987.
149. L. A. J. C. Ohlsson, "A Practice Driven Approach to Software Engineering-Education," *IEEE Transactions on Education*, vol. 38, pp. 291–295, 1995.
150. J. Boetje, "Foundational actions: teaching software engineering when time is tight," presented at the Conference on Innovation and Technology in Computer Science Education, New York, New York, USA, 2006, p. 285.
151. M. Gehrke, H. Giese, U. A. Nickel, J. Niere, M. Tichy, J. P. Wadsack, and A. Zündorf, "Reporting about industrial strength software engineering courses for undergraduates," presented at the International Conference on Software Engineering, New York, New York, USA, 2002, pp. 395–405.
152. Jain and B. Boehm, "SimVBSE: Developing a Game for Value-Based Software Engineering," presented at the Conference on Software Engineering Education and Training, 2006, pp. 103–114.
153. J. Tevis and K. Rouse, "Using industry-style software engineering and project management in a group project," *Journal of Computing Sciences in Colleges*, 2010.
154. R. F. Roggio, "A model for the software engineering capstone sequence using the Rational Unified Process®," presented at the Southeast regional conference, New York, New York, USA, 2006, p. 306.
155. K. Tian, K. Cooper, and K. Zhang, "Improving Software Engineering Education through Enhanced Practical Experiences," presented at the International Conference on Computer and Information Sciences, 2011, pp. 292–297.
156. J. Zhang and J. Li, "Teaching Software Engineering Using Case Study," *Proceedings of the 2010 International Conference on Biomedical Engineering and Computer Science (ICBECS)*, 2010, pp. 1–4.
157. Cordes and A. Parrish, "Incorporating re-use into a software engineering course with Ada," presented at the Washington Ada Symposium on Ada: Ada's role in software engineering, New York, New York, USA, 1993, pp. 109–114.
158. B. Demuth and D. Weigel, "Web Based Software Modeling Exercises in Large-Scale Software Engineering Courses," presented at the Conference on Software Engineering Education and Training, 2009, pp. 138–141.
159. K. Surendran, "Simulating a software engineering apprenticeship," *IEEE Software*, vol. 19, pp. 49—, 2002.
160. P. Teiniker, "A practical software engineering course with distributed teams," presented at the International Conference on Interactive Collaborative Learning, 2011, pp. 195–201.
161. Y. Sun and X. Liu, "Educating students by 'real-world' software engineering practice—Designing software engineering practice course," presented at the International Conference on Computer Science & Education, 2012.
162. N. Stankovic, "Single Development Project," presented at the Conference on Software Engineering Education and Training, 2008, pp. 171–178.
163. Baker, E. O. Navarro, and A. van der Hoek, "Problems and Programmers: an educational software engineering card game," presented at the International Conference on Software Engineering, 2003, pp. 614–619.
164. T. Rishel, "An innovative project structure for teaching software engineering," *Journal of Computing Sciences in Colleges*, 2012.
165. Y. Lester, "An Innovative Approach to Teaching an Undergraduate Software Engineering Course," presented at the International Conference on Software Engineering, 2008, pp. 301–306.
166. N. Stankovic, "Software engineering for undergraduates," presented at the International Conference on Software Engineering, 2006.
167. Q. Gates, N. Delgado, and O. Mondragon, "A structured approach for managing a practical software engineering course," presented at the *Frontiers in Education*, 2000, vol. 1.
168. Pletch and A. Agajanian, "A software engineering project that looks like the real world," *Journal of Computing Sciences in Colleges*, 2007.
169. P. Doerschuk, "Incorporating team software development and quality assurance in software engineering education," presented at the *Frontiers in Education*, 2004, pp. –480.
170. J. Burge and D. Troy, "Rising to the Challenge: Using Business-Oriented Case Studies in Software Engineering Education," presented at the Conference on Software Engineering Education and Training, 2006, pp. 43–50.
171. J. C. Dunlap, "Problem-based learning and self-efficacy: How a capstone course prepares students for a profession," *Educational Technology Research and Development*, 2005.
172. R. Kessler and N. Dykman, "Integrating traditional and agile processes in the classroom," *ACM SIGCSE Bulletin*, 2007.
173. J. S. Collofello, "Teaching practical software maintenance skills in a software engineering course," *ACM SIGCSE Bulletin*, 1989.

## REFERENCES

- [1] Petersen, F., R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering". *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, June 2008, pp. 71-80