



# Special Issue of ACM Transactions on Computing Education on Team Projects in Computing Education A Proposal

*Jürgen Börstler, Blekinge Institute of Technology, Sweden*  
*Thomas B. Hilburn, Embry-Riddle Aeronautical University, USA*

## I. Background: Why the computing education community will be interested

Modern software development projects have rapidly moved away from traditional “closed-shop models”, where a single collocated team develops, from scratch, a pre-specified piece of software for a known client according to a well-defined process (Shaw, 2000; Mead, 2009). Modern software is ubiquitous and typically just a part of a complex system that imposes complex inter-operability requirements based on changing platforms and technologies. Such software is often developed by distributed teams who follow different development processes and business models. This makes it more and more difficult to set up “real projects for real clients courses” (Klappholz et al., 2009), since there are a range of practical problems, which are difficult to simulate and/or control in an educational setting.

To prepare computing graduates for their professional careers, education must provide them with “real-life” experience. Walking students through all the software development phases is not sufficient to prepare graduates for professional work. There are many tasks beyond those of core development, for which students need training, like project management, team building, effort estimation, progress tracking, and communication. Communication, in particular, has become a critical issue, since teams in modern software development projects are often multidisciplinary and distributed over cultures and time zones. This makes course design, supervision, and assessment a challenge. How can we, for example, ensure fair individual assessment in a course, where distributed teams carry out different projects for different clients?

There is a large body of knowledge on issues and challenges pertaining to team project courses in computing education (Broman et al., 2012; Crnković et al., 2012; Fincher et al., 2001; Hilburn, 2002; Wikstrand & Börstler, 2006), problem- and project-based learning (Helle et al., 2006; Hmelo-Silver, 2004), and constructive alignment (Biggs, 1996; Reeves, 2006). Furthermore, there are accreditation regulations and international curriculum initiatives, like ABET (ABET) and CDIO (CDIO) that drive educational institutions to integrate the teaching of non-technical skills into their engineering and computing curricula. Reports on such integration would therefore be of significant interest to the computing education community, providing additional insight into the state-of-the-art regarding team projects in computing education. It would also offer support for research in the learning sciences and psychology.

There are numerous recent papers on specific issues related to team project courses. To our knowledge, though, there is no systematic treatment of this subject. There has been no comprehensive survey of the subject in more than 15 years.

## 2. Aims and scope of the issue

The aim of this special issue is to collect evidence-based information about team projects in computing education. The goal is twofold. On one hand, this information should help educators in improving the state-of-practice in software engineering education. On the other hand, it should bring forward research on various aspects related to the teaching and learning related to team-projects. We therefore solicit contributions covering one or more of the following issues:

- Team building and team dynamics, in particular regarding multidisciplinary teams
- Distributed development
- Development of non-technical skills
- Project process definition and tailoring
- Instructional design and constructive alignment
- Studio-based approaches, Makerspaces or other forms of project environments
- Real projects for real clients
- Curriculum issues and quality assurance
- Student-/team assessment
- Problem-based learning
- Cognitive and psychological aspects of team project courses
- Collaborative learning
- Methods and tools to support team project courses
- Motivation

Preference will be given to contributions that provide some form of empirical evaluation.

A preliminary one-page abstract of the paper is due January 2, 2014. Feedback will be provided to authors by February 1, regarding relevance of the proposed paper with respect to the Special Issue. Full papers submissions are due April 1, with publication expected in late 2014. Submissions must be done via Manuscript Central (<http://mc.manuscriptcentral.com/toce>). In the cover letter, please indicate that the paper is for the Special Issue on Team Projects in Computing Education. More information about the TOCE review criteria can be found on <http://toce.acm.org/authors.html>.

## 3. Links and resources

- Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2004): <http://www.acm.org/education>
- Graduate Reference Curriculum for Systems Engineering (GRCSE™): <http://www.bkcase.org/grcse>
- Graduate SE Curriculum Guidelines for Graduate Degree Programs in Software Engineering (GSWE2009): <http://www.acm.org/education>
- Team Software Process (TSP): <http://www.sei.cmu.edu/tsp>

#### 4. References

- ABET. Accreditation Criteria and Supporting Documents. Accreditation Board for Engineering and Technology, Inc. <http://www.abet.org/accreditation-criteria-policies-documents>.
- Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher education* 32(3), 347–364.
- Broman, D., Sandahl, K., & Abu Baker, M. (2011). The company approach to software engineering project courses. *IEEE Transactions on Education* 55(4), 445–452.
- CDIO. The CDIO Syllabus v2.0. CDIO Initiative (Conceive-Design-Implement-Operate). <http://www.cdio.org/framework-benefits/cdio-syllabus>.
- Crnković, I., Bosnić, I., & Žagar, M. (2012). Ten tips to succeed in global software engineering education. In *Proceedings of the 34<sup>th</sup> International Conference on Software Engineering*, 1225–1234.
- Fincher, S., Petre, M., & Clark, M. (Eds.). (2001). *Computer science project work: principles and pragmatics*. Springer.
- Hilburn, T.B., & Humphrey, W.S. (2002). Teaching teams. *IEEE Software*, 19(5), 72–77.
- Helle, L., Tynjälä, P., & Olkinuora, E. (2006). Project-based learning in post-secondary education – Theory, practice and rubber sling shots. *Higher Education* 51(2), 287–314.
- Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review* 16(3), 235–266.
- Klappholz, D., Almstrum, V.L., Modesit, K., Owen, C., & Johnson, A. (2009). A framework for success in real projects for real clients courses. *Software Engineering: Effective Teaching and Learning Approaches and Practices*. IGI-Global, 156–189.
- Mead, N.R. (2009). Software engineering education: How far we’ve come and how far we have to go. *Journal of Systems and Software* 82(4), 571–575.
- Reeves, T.C. (2006). How do you know they are learning? The importance of alignment in higher education. *International Journal of Learning Technology* 2(4), 294–309.
- Shaw, M. (2000). Software engineering education: a roadmap. *Proceedings of the 22<sup>nd</sup> International Conference on Software Engineering – The Future of Software Engineering*, 371–380.
- Wikstrand, G., & Börstler, J. (2006). Success Factors for Team Project Courses. *Proceedings of the 19<sup>th</sup> Conference on Software Engineering Education and Training*, 95–102.

#### 5. Brief biographies of the proposers

**Jürgen Börstler** is a professor of software engineering at Blekinge Institute of Technology, Sweden. His main research interests are in the areas requirements engineering, object-oriented modeling, software process improvement, and computer science education. He is a founding member of the Scandinavian Pedagogy of Programming Network (SPoP) established in fall 2004. He is a member of the editorial board of *Informatica Didactica*. He has co-organized numerous events in the area of computer science education and published several papers in computer science education related to introductory programming and software engineering.

**Thomas B. Hilburn** is a Professor Emeritus of Software Engineering at Embry-Riddle Aeronautical University and was a Visiting Scientist at the Software Engineering Institute, Carnegie-Mellon from 1997 – 2009. He has worked on software engineering development, research, and education projects with the FAA, General Electric, Lockheed-Martin, the Harris Corp, the MITRE Corporation, DOD, FIPSE, the SEI, the NSF, the ACM and the IEEE Computer Society. His current interests include software processes, case study teaching, software assurance, systems engineering, and curriculum development. In 2011, he was the recipient of the Nancy R. Mead Award for Excellence in Software Engineering Education. He also currently serves on the Computing Sciences Accreditation Board (CSAB).

## 6. Solicitation of contributions and potential contributors

The Call for Papers for this special issue will be spread through:

- Publication of the call in ACM inroads and ACM TOCE.
- Distribution through the SIGCSE, SEWORLD, and Informatics-Europe mailing lists.
- Distribution to computing departments throughout Australia, Asia, Europe, the Middle East, and the USA.

Contributions will furthermore be solicited through direct invitations send by the proposers to potential contributors. The proposers will invite authors of relevant conference papers (for conferences like CSEE&T, ICSE, ICER, ITiCSE, ACE, and SIGCSE) to submit extended versions.

## 7. Proposed timeline

The total lead time for the special issue will be approximately a full year starting Oct 1, 2013 and ending Sep 28, 2014 (see below).

Project Milestones	Deadline	Elapsed Time	Remark
Publish Call for Papers	Oct 1, 2013		See Section 6 for strategy.
<i>Abstracts due</i>	<i>Jan 2, 2014</i>	3 months	
Decisions on abstracts	Feb 1, 2014	30 days	
Reviewers recruited	Mar 1, 2014	28 days	
<i>Full papers due</i>	<i>Apr 1, 2014</i>	30 days	Authors have 8 weeks to prepare the full paper.
Papers to reviewers	Apr 8, 2014	1 week	
Reviews due	May 20, 2014	6 weeks	
Decisions on accept	Jun 1, 2014	2 weeks	
Notification of acceptance send out	Jun 9, 2014	1 week	
<i>Revised papers due</i>	<i>Jul 21, 2014</i>	6 weeks	
Final decisions on inclusion	Sep 1, 2014	6 weeks	
Editorial finished	Sep 14, 2014	2 weeks	Preparation can start before final acceptance.
<b>Issue complete</b>	<b>Sep 28, 2014</b>	2 weeks	