

## Team Projects in Computing Education \*

JÜRGEN BÖRSTLER, Blekinge Institute of Technology

THOMAS B. HILBURN, Embry-Riddle Aeronautical University

Team projects are a way to expose students to conflicting project objectives, and “[t]here should be a strong real-world element . . . to ensure that the experience is realistic” [ACM/IEEE-CS 2015b]. Team projects provide an opportunity for students to put their education into practice and prepare them for their professional careers. The aim of this special issue, and the previous one, is to collect and share evidence about the state of practice of team projects in computing education and to help educators in designing and running team projects. The articles presented in the present issue cover the following topics: real projects for real clients, open source projects, multidisciplinary team projects, student and team assessment, and cognitive and psychological aspects of team projects.

CCS Concepts: • **Social and professional topics** → **Computing education programs**

Additional Key Words and Phrases: Team projects

### ACM Reference Format:

Jürgen Börstler and Thomas B. Hilburn. 2016. Team projects in computing education. *ACM Trans. Comput. Educ.* 16, 2, Article 4 (March 2016), 4 pages.

DOI: <http://dx.doi.org/10.1145/2808192>

## 1. INTRODUCTION

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.

To prepare computing graduates for professional careers, their education must provide them with “real-life” experience. Simply lecturing about and discussing all of the software development phases is not sufficient preparation for a professional career. There are many tasks beyond those of core software development for which students need training: project management, team building, software estimation and planning, 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. Global and distributed development also make it more difficult to set up “real projects for real clients” courses [Klappholz et al. 2009], as there are a range of practical problems that are difficult to simulate and/or control in an educational setting.

---

Authors’ addresses: J. Börstler, Software Engineering Research and Education Lab Sweden, Blekinge Institute of Technology, Campus Gräsvik, 371 79 Karlskrona, Sweden; T. B. Hilburn, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114-3900, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

2016 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 1946-6226/2016/03-ART4 \$15.00

DOI: <http://dx.doi.org/10.1145/2808192>

There is a large body of knowledge on issues and challenges pertaining to team project courses in computing education [Crnković et al. 2012; Ellis et al. 2009; Fincher et al. 2001; Hilburn and Humphrey 2002; Wikstrand and Börstler 2006] and problem- and project-based learning in general [Helle et al. 2006; Hmelo-Silver 2004]. Furthermore, there are accreditation regulations and international curriculum initiatives, like ABET [2015], CDIO [2011], and ACM/IEEE-CS [2015a] that drive educational institutions to integrate the teaching of nontechnical skills into their engineering and computing curricula.

This and the subsequent issue present the experience and research of educators, which we believe can help those who want to improve the delivery of team project courses, and which supports the overall goal of preparing graduates for professional practice.

## 2. KEY AREAS OF INTEREST

The aim of this special issue and the previous one is to share evidence-based practices that have been applied to team projects in computing education. The goal is twofold. On one hand, this information should help educators improve the state of practice in computing education. On the other hand, it will bring forward research on various aspects related to the teaching and learning about team projects.

The call for papers for this special issue required an initial submission of an abstract. In total, a record number of 69 abstracts were submitted from all over the world: Australia, Canada, China, Egypt, England, Estonia, Finland, Germany, Greece, India, Israel, Italy, Jamaica, Macedonia, Netherlands, Norway, Poland, South Africa, Spain, Sweden, and the United States.

The editors reviewed the abstracts submitted and invited 19 authors to submit a full paper. A team of 26 reviewers reviewed the submitted papers, and based on the reviews, nine were finally accepted for publication in this special issue. Papers in the following categories were accepted:

- Multidisciplinary team projects
- Studio-based approaches
- Cognitive and psychological aspects
- Student and team assessment
- Team building and team dynamics
- Collaborative learning and methods and tools to support team project courses
- Real projects for real clients
- Open source projects.

## 3. ARTICLES IN THIS ISSUE

The articles in this issue and a preceding issue [Börstler and Hilburn 2015] range over the categories listed previously. What follows is a short description of each article.

### 3.1. Studio-Based Approaches

In the first article of this special issue, “Interdisciplinary Projects in the Academic Studio,” Gestwicki, and McNely describe the academic studio model for interdisciplinary, undergraduate, project-oriented education. The academic studio discussed in this article integrates agile software development practice (using Scrum), project-oriented pedagogy, and sociocultural cognition theories.

The authors’ model emerged from design-based research, which investigated the relationship of fun, games, and learning through the development of educational video games. Their analysis of the research reaches the conclusion that the academic studio model is beneficial to student learning and faculty development.

### 3.2. Collaborative Learning and Methods and Tools to Support Team Project Courses

In “A Method to Analyze Computer Science Students’ Teamwork in Online Collaborative Learning Environments,” Vivian, Falkner, Falkner, and Tarmazdi adopted an in-depth case study approach to analyze teamwork discussions to determine if the frameworks, which they developed for the study, provided insight into student teamwork behavior. They also sought to determine if the task encouraged students to collaborate and share knowledge, and self-adopt teamwork roles.

Analysis revealed the identification of both active and cohesive teams, disengaged students, and particular roles, skills, and behaviors that were lacking at an individual and team level. The findings demonstrate the value in the detailed measurement and analysis of online teamwork. The authors state the need for automated measures that provide real-time feedback to assist educators in the fair and efficient assessment of teamwork. They present a prototype system and recommendations for automated teamwork analysis tools.

### 3.3. Team Building and Team Dynamics

In “Measuring and Understanding Team Development by Capturing Enthusiasm and Skill Levels,” Largent argues that to prepare graduates for today’s work environment, they must be immersed in positive (and perhaps negative) small group experiences in their courses, which will in turn provide a basic understanding of how teams form and develop over time. The article describes a research study that explores how software development teams form and interact in a computer science college capstone course setting. The focus of the research was on the experiences of computer science college course teams as compared and contrasted to the Tuckman small group development model.

The author analyzed data collected over 5 years from 51 project teams. The data shows patterns similar to that of Tuckman’s model. The article proposes Tuckman’s model as an effective tool to teach teamwork and monitor team development, and presents a simple conceptualization of the model that can be captured in two data points: enthusiasm and skill level. The author describes a mini-curriculum that can be used to introduce students to Tuckman’s model and provide them with insight into which leadership style works best in each of the development stages.

### 3.4. Cognitive and Psychological Aspects

In “Exploration of Participation in Student Software Engineering Teams,” Marshall, Pieterse, Thompson, and Venter describe an experiential learning exercise that was designed to teach the software engineering process in conjunction with teamwork skills. The underlying teaching strategy applied in the exercise maximizes risks to provide maximal experiential learning opportunities. Students are expected to work in fairly large, yet short-lived, instructor-assigned teams to complete software engineering tasks. After undergoing the exercise, students form self-selected teams for their capstone projects.

The authors determine and report on the influence the teaching exercise had on the formation of teams for the capstone project. The article contends that by analyzing data provided by students through regular peer reviews, insight is gained into the teams’ dynamics as well as to what extent the members contributed to the team effort. In conclusion, the authors present their observations from the analysis of team compositions, team types, and team migrations, and provide directions for future work and collaboration.

#### 4. DIRECTIONS FOR FUTURE RESEARCH

The articles in this issue motivate and support research in the following ways:

- Exploring the differences in team role adoption and group cohesion between different year levels, disciplines, and locations
- Understanding the interplay between the different participatory levels of students
- Evaluating how the Tuckman model may help educators distinguish between normal difficulties in team development (e.g., in the storming stage) and team difficulties
- Understanding the effects of the academic studio model on the capability of students to develop multidisciplinary collaboration skills.

#### ACKNOWLEDGMENTS

We would like to thank the following reviewers who supported us in selecting several interesting and high-quality contributions: Carl Alphonse, Mark Ardis, Tapio Auvinen, Don Bagert, Ellen Francine Barbosa, Nauman Bin Ali, Tony Clear, Tony Cowling, Stephen Frezza, Keith A. Garfield, Dahai Guo, Shafagh Jafer, Ville Karavirta, Anna Koufakou, Timothy Lethbridge, Stephani Ludi, Eduardo Miranda, Julia Prior, Damith C. Rajapakse, Tom Reichlmayr, Salamah I Salamah, Carol Sledge, Sean Thorpe, Jim Vallino, Arto Vihavainen, and Greger Wikstrand. Without their expertise and time, developing this special issue would have been an impossible task. We would also like to thank the editors of TOCE, Josh Tenenberg and Robert McCartney, for their guidance and support in developing this special issue.

#### REFERENCES

- ABET. 2015. Accreditation Criteria and Supporting Docs. Retrieved January 19, 2016, from <http://www.abet.org/accreditation/accreditation-criteria/>.
- ACM/IEEE-CS. 2015a. Curricula Recommendations. Retrieved January 19, 2016, from <http://www.acm.org/education/curricula-recommendations>.
- ACM/IEEE-CS. 2015b. *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. Draft Report Pending Approval.
- Jürgen Börstler and Thomas B. Hilburn. 2015. Team projects in computing education. *ACM Transactions on Computing Education* 15, 4.
- CDIO. 2011. The CDIO Syllabus v2.0. Retrieved January 19, 2016, from <http://www.cdio.org/framework-benefits/cdio-syllabus/>.
- Ivica Crnković, Ivana Bosnić, and Mario Zagat. 2012. Ten tips to succeed in global software engineering education. In *Proceedings of the 34th International Conference on Software Engineering*. 1225–1234.
- J. C. Heidi Ellis, Steven A. Demurjian, and J. Fernando Naveda. 2009. *Software Engineering: Effective Teaching and Learning Approaches and Practices*. IGI Global, Hershey, NY.
- Sally Fincher, Marian Petre, and Martyn Clark. 2001. *Computer Science Project Work: Principles and Pragmatics*. Springer Science & Business Media, London, UK.
- Laura Helle, Päivi Tynjälä, and Erkki Olkinuora. 2006. Project-based learning in post-secondary education—theory, practice and rubber sling shots. *Higher Education* 51, 2, 287–314.
- Thomas B. Hilburn and Watts S. Humphrey. 2002. Teaching Teamwork. *IEEE Software* 19, 5, 72–77.
- Cindy E. Hmelo-Silver. 2004. Problem-based learning: What and how do students learn? *Educational Psychology Review* 16, 3, 235–266.
- David Klappholz, Vicki L. Almstrum, Ken Modesit, Cherr Owen, Allan Johnson Allen, and Steven J. Condly. 2009. A framework for success in real projects for real clients courses. *Software Engineering: Effective Teaching and Learning Approaches and Practices: Effective Teaching and Learning Approaches and Practices*, H. J. C. Ellis, S. A. Demurjian, and J. Fernando Naveda (Eds.). IGI Global, Hershey, NY, 157–189.
- Nancy R. Mead. 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.
- Mary Shaw. 2000. Software engineering education: A roadmap. In *Proceedings of the 22nd International Conference on Software Engineering—The Future of Software Engineering*. 371–380.
- Greger Wikstrand and Jürgen Börstler. 2006. Success factors for team project courses. In *Proceedings of the 19th Conference on Software Engineering Education and Training*. 95–102.

Received July 2015; accepted July 2015