Industry Agile Practices in Large-scale Capstone Projects

Jean-Guy Schneider, Peter W. Eklund, Kevin Lee, Feifei Chen, Andrew Cain, Mohamed Abdelrazek
School of Information Technology
Deakin University, Geelong, Australia
{jeanguy.schneider,peter.eklund,kevin.lee,feifei.chen,andrew.cain,mohamed.abdelrazek}@deakin.edu.au

ABSTRACT

To give students as authentic learning experience as possible, many software-focused degrees incorporate team-based capstone projects in the final year of study. Designing capstone projects, however, is not a trivial undertaking, and a number of constraints need to be considered, especially when it comes to defining learning outcomes, choosing clients and projects, providing guidance to students, creating an effective project "support infrastructure", and measuring student outcomes. To address these challenges, we propose a novel, scalable model for managing capstone projects, called ACE, that adapts Spotify's *Squads and Tribes* organization to an educational setting. We present our motivation, the key components of the model, its adoption, and refer to preliminary observations.

KEYWORDS

Software Engineering Education, Agile Software Development, Capstone Projects

1 INTRODUCTION

To help students transition from an educational environment to a professional workplace, and better prepare students for the lifelong learning required in contemporary workplaces, educational institutions are increasingly focusing on project-based learning (PBL). PBL prepares students to solve real-world, ill-structured problems, improving students outcomes in conceptual understanding, problem-solving and meta-cognitive skills [5], giving them insights into technology, product development, and teamwork [3].

In a "typical" Software Engineering curriculum, PBL focuses on the latter years of Bachelor's and/or Masters degrees, in the form of so-called "capstone projects." In line with industry expectations, these projects increasingly adopt agile methodologies, and project teams are expected to produce working prototypes across a number of iterations over a teaching period. Although this sounds straightforward, adopting an agile approach to Software Engineering education is a non-trivial undertaking that requires considerable care, not only devising the core elements of capstone projects, but also providing students with adequate scaffolding to succeed with agile approaches.

ICSE '20 Companion, October 5–11, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7122-3/20/05.
https://doi.org/10.1145/3377812.3390794

Inspired by state-of-the-art practices documented by relevant literature [1], as well as extensive personal experience of the authors [7, 8], a novel, scalable model for managing capstone projects in education, called ACE, is proposed. The model: (i) implements learning outcomes we expect students to demonstrate on completion of their capstone project; (ii) adapts Spotify's *Squads and Tribes* model [2] to an educational setting and; (iii) defines a teaching period timeline that provides students with a scaffolding to succeed in an agile setting.

We summarize our motivation behind the model, illustrate its key components, and refer to preliminary observations of its adoption in a Post-Graduate (PG) setting. The interested reader is referred to the full paper [6] for details.

2 MOTIVATION

PBL is a key cornerstone of achieving *Graduate Learning Outcomes* (GLOs)[1] for an ICT-degree, not only in relation to discipline-specific technical skills, but also to develop teamwork, communication, and project management skills. Further, GLOs must reflect industry expectations of what an ICT-graduate should "look like" when they complete their studies. Therefore, it is important to appropriately design (*i.e.* allowing students to demonstrate their GLOs) as well as effectively resource and support capstone projects in order to achieve the best possible experience for all stakeholders.

The following introduces high-level goals for supporting successful capstone units from experience – (elaborated in [6]):

Motivation 1: Industry relevant experience
Motivation 2: Authentic learning experience
Motivation 3: Continuity of learning experience
Motivation 4: A successful client experience
Motivation 5: Scalability of approach
Motivation 6: Effective support of students

There is a clear need for students to have an *Industry relevant, authentic learning experience*, allowing them to work on challenging projects with obvious societal benefits. Evidence shows that capstone projects with an industry focus benefit both the student learning experience and project outcomes alike [3, 4]. There is also a need to provide students with *continuity* in their learning experience – capstone projects often run across two (or more) teaching periods – as well as effective support in order to enhance the chances of successful outcomes, both for students and project clients alike. Finally, to effectively support capstone projects with large cohorts of hundreds (to thousands) of students, it is important to have the *Scalability of approach* as a consideration when designing the capstone experience.

[1] Graduate Learning Outcomes are comparable the *Knowledge Areas* in the "Computing Curriculum – Software Engineering", *cf.* http://sites.computer.org/ccse/know/FinalDraft.pdf

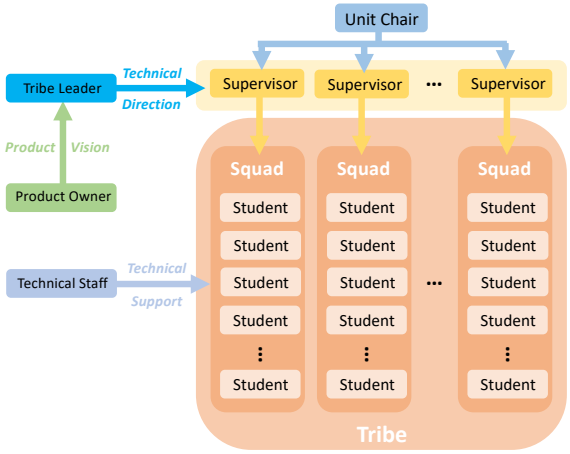

Figure 1: Tribes and Squads of proposed capstone model.

3 ACE: AGILE CAPSTONE IN EDUCATION

Figure 1 illustrates the key components of our capstone project model. The model is inspired by Spotify's "Tribes and Squads" [2], adjusted to an educational environment.

The overarching building block of the model is a **Product Tribe** where a large, *long-term* product undergoes *enhancements* over a number of teaching periods. The vision of the Product Tribe is set by a **Product Owner** who is an industry representative. A **Tribe Leader** is an academic staff member with a specific interest in the project, who translates the vision into manageable *work packages* for each teaching period. Within a Product Tribe there are a number of **Squads** (teams of students) responsible to deliver a set of work packages over the duration of a teaching period. Depending on what each Squad delivers, the vision of the Product Tribe is reviewed and a new set of work packages are identified for each teaching period.

The Tribe Leader acts as the gatekeeper between students and Product Owner and helps streamline communication, shielding the external Product Owner from the challenges students face with externally-facing communication, such as relevancy or frequency, and hence improves the experience of Product Owner. Each squad has a dedicated **Project Supervisor** (an academic staff member) who oversees the students' work, mentors them in project activities, and engages in weekly "stand-up" meetings. Squads and their supervisor are organized around Microsoft Teams groupware that is used for communication, collaboration and file exchange, respectively.

At our institution, students are required to complete two capstone project units. To minimize effort in inducting students into a Product Tribe, Squads are comprised of (a roughly equal number of) students from *both* capstone units. We refer to students in the first capstone unit as *Junior* students and students in their second unit as *Senior* students. The Senior students are mainly responsible to drive a Squad forward while the Junior students are there to learn about the Product Tribe, and increasingly support Senior students in working on deliverables. The Squads of a Product Tribe are reset at the start of each teaching period and Senior students do not necessarily work with the same peers as before. This gives students a team-work experience similar to many real-world workplaces, where the composition of work teams changes routinely.

Two **Unit Chairs** with overall responsibility for grading and assessment coordination oversee all activities during a teaching period. They are also responsible to moderate expectations across Squads to ensure that all students get an equal learning experience and are assessed as fairly and transparently as possible.

To provide students with a guiding framework to succeed in an agile environment, we divided the teaching period into four 3-week time periods (or *iterations*), each of which has clearly defined outcomes Squads are expected to meet. We also introduced a *physical collaboration work-space* (equipped with a mix of meeting rooms, wall-to-wall whiteboards, large screens, workstations etc.) with exclusive access for the capstone project students and a co-located help-hub with **Technical Advisers** that provide Squads technical support and also run workshops on specific technologies.

4 EARLY OBSERVATIONS

The proposed model was deployed in 2019 across the two PG capstone project units within our School, across a total of 16 Product Tribes, catering for approx. 300–350 students per teaching period. Hence we achieved scalability of the model as we did not have to cater for this number of students in previous teaching periods.

All students and Project Supervisors were surveyed at the end of the second teaching period. The responses were predominantly positive, both from students and supervisors alike, giving us confidence that we are achieving the objectives listed in Section 2. Most notable are that the majority of respondents felt that (i) the capstone experience created an environment similar to what is expected in industry and (ii) the capstone experience improved students' IT skills. Further, most students felt positively about the mentoring of Junior students by their Senior counterparts, but we identified aspects of the mentoring that requires further refinement. There is also anecdotal evidence that the students felt more confident finding employment in the ICT sector after their capstone experience.

REFERENCES

[1] Malek Al-Zewairi, Mariam Biltawi, Wael Etaiwi, and Adnan Shaout. 2017. Agile software development methodologies: survey of surveys. *Journal of Computer and Communications* 5, 05 (2017), 74–97.
[2] Mashal Alqudah and Rozilawati Razali. 2016. A Review of Scaling Agile Methods in Large Software Development. *International Journal on Advanced Science, Engineering and Information Technology* 6, 6 (2016), 828–837.
[3] Bernd Bruegge, Stephan Krusche, and Lukas Alperowitz. 2015. Software Engineering Project Courses with Industrial Clients. *ACM Transactions on Computing Education (TOCE)* 15, 4 (Dec. 2015), 17:1–17:31.
[4] Bahram Nassersharif and Linda Ann Riley. 2012. Some best practices in industry-sponsored capstone design projects. In *2012 Capstone Design Conference*. Champaign-Urbana, Illinois, USA.
[5] Michael J. Prince and Richard M. Felder. 2006. Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. *Journal of Engineering Education* 95, 2 (2006), 123–138.
[6] Jean-Guy Schneider, Peter W. Eklund, Kevin Lee, Feifei Chen, Andrew Cain, and Mohamed Abdelrazek. 2020. Adopting Industry Agile Practices in Large-scale Capstone Education. In *Proceedings of IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. ACM, Seoul, South Korea.
[7] Jean-Guy Schneider and Lorraine Johnston. 2005. eXtreme Programming – Helpful or Harmful in Educating Undergraduates? *Journal of Systems and Software* 74, 2 (Jan. 2005), 121–132.
[8] Jean-Guy Schneider and Rajesh Vasa. 2006. Agile Practices in Software Development – Experiences from Student Projects. In *Proceedings of the 17th Australian Software Engineering Conference*. Sydney, Australia, 401–410.