Teaching a Globally Distributed Project Course Using Scrum Practices

Daniela Damian*, Casper Lassenius†, Maria Paasivaara†, Arber Borici* and Adrian Schröter*

*SEGAL
University of Victoria, Victoria, BC, Canada
Email: danielad@cs.uvic.edu, borici@uvic.ca, schadr@acm.org
†Software Process Research Group
Aalto University, Espoo, Finland
Email: casper.lassenius@aalto.fi, maria.paasivaara@tkk.fi

Abstract—This paper describes the goals, design and initial challenges encountered in teaching a globally distributed software development course in collaboration between the University of Victoria, Canada and Aalto University, Finland. The project-driven collaboration course involved 16 students in Canada and nine students in Finland, divided into three globally distributed Scrum teams working on the same project. The teams worked on extending Agilefant, an open-source backlog management system, in direct interaction with its product owner. The collaborative development is based on the Scrum methodology. We describe how the Scrum methodology was implemented, and adapted to work in a distributed environment, as well as the infrastructure used to support collaboration, e.g. local war-rooms, and multiple communication tools. We conclude the paper with describing initial challenges encountered, including cultural, semester, course and curriculum differences, as well as technical and time-zone issues.

Keywords—software engineering education; distributed Scrum; agile development; global software engineering

I. INTRODUCTION

Modern software engineering is global and often done in globally distributed teams [1]. Changing requirements and demands of the software engineering industry must be addressed in modern curricula to respond to such demands by adding or improving our graduate students’ skills [2].

Agile methods, like Scrum [3] and XP [4], are increasingly being adopted by software organizations all over the world [5]. Originally, agile methods, were designed for the use of single small teams, members of which work face-to-face preferably in team rooms. Although agile development per se seems to be a poor fit in distributed development, due to its focus on face-to-face communication, it is increasingly being adopted also in such challenging settings [6]–[10]. However, scaling agile practices to large and/or distributed projects involves several challenges, such as coordination and communication between the teams, and architectural challenges [11]. Despite the challenges several companies have already applied agile practices to their large distributed projects and a few books (e.g. [11], [12]) have been published describing practices how to scale agile methods.

This paper presents the set up and the first experiences and lessons learned from a globally distributed software project course taught in collaboration between the University of Victoria, Canada and Aalto University, Finland. The paper is structured as follows: Section II presents an overview of the course, Section III discusses the development project, Section IV describes the collaboration infrastructure, and Section V presents initial challenges encountered. Finally, the conclusions are presented in Section VI.

II. COURSE OVERVIEW

A. Learning Objectives and Outcomes

The course has three main objectives:

1) To teach global software engineering (GSE) skills and strategies by emulating a modern real-world GSE environment through the use of agile practices that involve significant inter-team and inter-site communication;

2) To teach and expose students to various collaborative (CSCW) technologies that provide infrastructural as well as cognitive support for effective communication in global teams;

3) To expose students to cultural differences in a GSD context.

B. Course Design

We arranged the course in collaboration between the University of Victoria, Canada and Aalto University, Finland.

1) Canadian Course: At the University of Victoria the course began in January 2012 and is being run in a 4-month traditional lecture-assignments-project structure in which students are required to participate in a term-long project as well as attend lectures and participate in in-class as well as online discussions about the material taught in class.

The main learning outcomes for the Canadian course could be succinctly summarized as follows:

• To work in a real GSD project with a real client.
• To plan work with the architect and the client (product owner).
• To follow an agile development process with frequent iterations. The iterations, also referred to as sprints, occur every two weeks preceded by a retrospective
account of the previous sprint as well as by planning the next sprint.
• To become involved in an existing project with ongoing global development.
• To work with a collaborative development environment designed for global software teams and learn new collaborative technologies.
• To develop skills of teamwork, work estimation and planning, and effective communication across large time zone differences, all in a global context.
• To overcome barriers of inter-cultural differences by working within multicultural teams.

Following the GSE instructional design framework [2] the elements of the course design included: an initial set of material- and discussion-based lectures, training on the Agile methodologies and practices, ongoing selected reading and blogging on readings or the experience in the course, and an ongoing project work in global teams.

In the Canadian course, 50% of the assessment comes from the project component of the course, whereas the rest is divided between lectures, reading and blogging, and final papers. At first, students were required to attend a set of preliminary lectures on GSD while also training on Agile development processes. The lectures, taught by the course instructor, covered the motivation for going global and an ongoing project work in global teams.

The course had a major project component as the main pedagogical tool in effecting the learning of GSD skills. Besides emulating a real GSD environment, one of the main drivers of the course was to have students apprehend the challenges involved as well as coordination and communication needs that arise in global software engineering. In turn, this would raise awareness of, as well as fostering the development of the necessary coping skills to solve GSE problems as well as to overcome cultural barriers by learning how to communicate effectively. In that respect, developing international teamwork skills and establishing a remote client-developer work relationship were the two main drivers in the design of the course project, which we describe next.

1) Finnish Course: At Aalto University in Finland, the course is run as a capstone project in which students are required to attend no lectures but complete a 5-month long project in which to apply the software engineering skills acquired in their software engineering program thus far. Software engineering students are required to take this capstone course twice during their studies: first as coders in their B.Sc. studies, then as software engineering experts during their M.Sc. level studies. The goal of the project course is to simulate real software projects; thus all projects have external customers, the majority of which are from industry. Annually, the course attracts about 100–120 students, executing about 20 projects, which is about half of the number of topics offered. Projects are implemented using an iterative process in teams of 7–10 students. This year, one student team was offered the option to learn global software engineering as part of their project work.

The Finnish course started in September 2011, as shown in the timeline in Figure 1. During the first semester of the Finnish course, the Aalto students worked in a collocated fashion. The Finnish course has one instructor, and each team is assigned a staff mentor who supports the team during the project. While the Finnish course prescribes a fairly loose process framework, the Finnish team used the Scrum development methodology [3] from the outset of their project. The reason for this was the intention to later use distributed Scrum in collaboration with students from Canada. However, at this stage, in-tram communication, as well as communication with the product owner was mainly done face-to-face. The team was allocated a “war room” at the University, in which students typically spent two days each week working on the project.

As evident from the picture, the Finnish course officially ends at the end of February, while the joint activities with Canada continue until early April. Thus, special arrangements with the students in Finland granting them extra credits for continuing the course until April were made.

III. Project Overview

After the Canadians joined, the project had a total of 25 PhD, Master’s and undergraduate students: 16 in Canada and nine in Finland, all working on a common goal: to extend Agilefant, a free backlog management system developed at Aalto University.

A. Teams and Roles

We split the students into three distributed Scrum teams, each consisting of 7–8 members. Each team comprises of 4–5 students from Canada and 3 students from Finland, see Figure 2. The product owner (PO), a staff member at Aalto University, was located in Finland. As the Finnish students had worked on the project for several months before the Canadian students joined, one of them was selected to function as the joint Scrum Master (SM) for all teams, as shown in the figure.

B. Project Initiation

Using lessons learned in the field of Global Software Engineering, e.g. [6], the Canadian instructor visited and spent one day with the whole Finnish team in December 2011 before the Canadian course started, to give the Canadian site a “face”.

In January 2012, when the Canadian course commenced, the Aalto instructor and a subset of the Finnish team comprising of the product owner (PO) and two students

1See https://noppa.aalto.fi/noppa/kurssi/t-76.4115/
2See http://www.agilefant.org/.
with extensive technical knowledge of the project and agile practices visited the Canadian site for nine days. During the visit, Scrum training, as well as technical training on Agilefant was given. The Canadians also performed one highly coached six-day "mini-sprint" to familiarize themselves with both the Scrum practices and the Agilefant codebase. At the end of the visit, the development of new Agilefant features resumed with a new approach to software development—that of agile GSD.

C. Development Process

The development process is a fairly strict implementation of Scrum, adopted for distributed projects, using two week long sprints.

At the beginning of each sprint, the teams do synchronous sprint planning. At the beginning of the sprint planning, the product owner presents an overview of the user stories to be developed in the sprint, as well as an initial allocation of user stories to the three teams in a joint videoconference session. Then, the teams split into three separate videoconference sessions. The product owner visits each team and explains the user stories in more detail, to give the team a basis for story-point estimation. The teams estimate the stories, and perform an initial task breakdown. Immediately after their team-specific sessions, when all teams have made their estimations and task breakdowns, the teams communicate their sprint estimates to the product owner and each other in another joint videoconference session. All planning activities take place synchronously, with the team members participating either from one of the war rooms (see Section IV), or using their own computers in whatever location they happen to be.

During the sprints, teams have two weekly standup meetings ("daily scrums"). Since the students work 10–12h/week on the project, this frequency for standups has been determined adequate. The standup meetings are scheduled by each team separately, and typically takes place early morning in one site, which is night time at the other site. For example, one team has one of their hangouts at 9:00 AM Finnish time, which is 11:00 PM in Canada.

At the end of the sprint, teams do a joint demonstration to the product owner and each other, as well as perform team retrospectives. For practical reasons, demonstrations, retrospectives and sprint planning typically all take place in a single, time boxed session that lasts about 2.5 hours.

D. Development Tools

The teams are exposed to several tools. Development is done using Eclipse is the integrated development environment (IDE) in conjunction with IBM’s RTC Jazz collaborative platform. Because the existing Agilefant repositories are hosted on Github, students were asked to connect their IDE via eGit. Moreover, in order to provide and maintain cognitive support for coordination needs, students use ProxiScientia, a tool developed at UVic that visualizes coordination requirements in real time.

3 See http://www.github.com/.
IV. THE COLLABORATION INFRASTRUCTURE

The students use several forms of intra-team, inter-team, and product owner-team collaboration: synchronous and asynchronous, in a ‘virtual’ war room, where communication has occurred both face to face as well as via videoconferencing, and distributed in pairs or in larger groups. Through this process students are establishing trust, common ground, and shared knowledge with the product owner as well as with local and remote colleagues.

To provide dedicated support to the global collaboration and communication between the teams and the client, a solid infrastructure was made available to the teams. The infrastructure comprises a wide range of tools and interfaces for synchronous and asynchronous interactions. For instance, students have exploited Google+ Hangouts for synchronous, multi-feed videoconferencing; Skype for voice calls; the Internet Relay Chat (IRC) instant messaging system, and have made extensive use of email through dedicated mailing lists. It is interesting to note the leveraging by teams of Google+ – a relatively recent service – to hold the standup meetings and iteration planning. Due to its multiple video feed capability, the Google+ Hangout feature eschews the distress about locality because individual team members can choose to participate wherever they wish from.

Moreover, project members have the option to use physical war rooms both in Canada and Finland. During standup meetings, demos, sprint planning and retrospectives, students can participate either from one of these physical locations, or from their homes using Google hangout. Standup meetings last for about 15 minutes; thereafter, team members usually choose to hang out informally and discuss project-related issues or strategies for the given iteration. When demoeing the completed stories to the product owner, assessing the current iteration retrospectively and planning for the next iteration, interactions occur synchronously for about 2 hours.

The Canadian war room contains three separate offices serving for private or ad-hoc meetings and other development-related purposes. For instance, pairs of a team may choose to work on assigned user stories at one of the offices. Also, when used as a scrum room, the equipment the teams are able to leverage include several multi-feed cameras as well as five smart boards which act as videoconferencing interfaces as well.

The Finnish was room is an open office space with several computers, but no dedicated equipment. When participating in distributed sessions from this location, students gather in small groups in different parts of the room, contacting their Canadian counterparts from the available computers or their own laptops.

V. ENCOUNTERED CHALLENGES

We observed a number of challenges that the students have encountered in the first six weeks of this course. They include cultural differences, technical difficulties, time-zone differences due to the geographical distance between British Columbia and Finland, and incongruent curricular activities between the two participating universities. We briefly describe these challenges here.

A. Cultural Differences

A major cultural difference that we observed is the way thoughts, ideas, and opinions are conveyed. The Finnish team members are generally more direct in discourses with the Canadian counterparts, which is an unusual cultural trait amongst the latter. However, Canadian students tend to be more open about a spectrum of issues than are the Finnish students. Communication has occurred in English and no language-related issues have been acknowledged, despite the fact that there are non-native speakers on both sites.

B. Technical Challenges

Due to the ongoing development by a variety of developers, and a strict focus on developing new functionality, the Agilefant system is poorly documented. This has been a source of frustration and challenge for some, if not all, both Canadian and Finnish students. While the Finnish students have both had more time to get to know the product, as well as have the benefit of having one person—the Scrum Master—with a long history of Agilefant development, a significant amount of their time has been spent learning the existing code.

The situation is more difficult for the Canadian students, who have less experience with the system than the Finnish one, and no on-site system expert. Due to the time difference, getting technical answers quickly from the Finnish team members is a real challenge that in some cases have significantly slowed their progress.

While frustrating to students, the challenge of having to work on more or less undocumented code is not uncommon in industry, either.

C. Time-Zone Differences

The time-zone difference between British Columbia and Finland is ten hours. This poses a challenge in terms of organizing an effective schedule for holding synchronous standup meetings, iteration retrospectives and planning, as well as for collaborating and communicating across sites. In particular, students in both countries had to make additional effort in coping with the pain of large time zone difference. Canadians made an effort to allot extra time beyond the regular class hours. Students in Finland also had to set aside some time to enable this collaboration. For instance, to accommodate the time-zone differences, one wave of standup meetings occurs at 10:30 PM PST, and the other during weekdays early in the morning in Canada (PST), which is early evening in Finland (EET).
Nonetheless, we have observed that team members have been leveraging the aforementioned collaboration infrastructure to set up and hold as many meetings as necessary. For example, to demonstrate completed user stories to the product owner, individual pairs establish communication with the product owner at irregular times during business days as well as on weekends.

D. Semester, Course and Curriculum Differences

The format of the course at the University of Victoria is different from that at Aalto University in several aspects. First, due to semestral differences. The course at Aalto begun in September 2011, whereas the Canadian students joined it in January 2012.

Second, the Canadian course is divided into lectures, reading and blogging on readings, and the project work, whereas the Aalto course is a capstone project course where time is dedicated solely to project work, with no lectures. To prepare the Finnish students for the GSE aspects of the course, the Canadian instructor visited Finland in December, and the Finnish instructor provides additional coaching to the project team, outside the normal curriculum.

Third, the time dedicated to the actual project work is imbalanced between the Canadian team members and the Finnish ones: the latter is dedicating, on average, 10 hours a week to the project (and no other activities are required), whereas the former are normally expected to spend 8-12 hours a week for the entire coursework (including classes and required readings).

While these curricular discrepancies have been sufficiently challenging for both sides, students seem to have been making an effort to induce an optimal flow of activities.

VI. CONCLUSIONS

In this paper we described the goals, course setup as well as the infrastructure and initial challenges encountered in a joint GSE course between Aalto University, Finland and the University of Victoria, Canada.

Despite the challenges discussed, the course is progressing well, and students comment positively on both the experience of developing a "real product", as well as on the global development context.

At the time of writing, the course is still in progress. The course is also a source for research data on tool support and collaboration in GSE projects. We look forward to report our full experiences, as well as our experimental results in a future paper.

REFERENCES