

Accredited Software Engineering Capstone projects adapting to agile offshoring

Sita Ramakrishnan

Monash University, Faculty of IT, Clayton, Victoria, Australia

sitar@infotech.monash.edu.au

Abstract

Software Engineering capstone projects have been running successfully since 2002 for the final year software engineering (SE) students of the Bachelor of Software Engineering (BSE) Program at Monash University, accredited by Engineers Australia and Australian Computer Society. In this paper, we report on our recent effort towards setting up an environment for capstone projects in offshore settings with distributed teams. We discuss our customised Monash University Software Engineering (MUSE) service-oriented portal which was set up primarily for SE capstone team projects in 2004. We are adapting our approach with a combination of open-source and commercial tools under academic licence for developing and deploying these projects effectively with distributed teams with an offshoring component.

1. Motivation

Offshore software development (OSD) is a form of outsourcing which has to take on board the complexity of developing software systems when the contracted consultants reside in different countries [1]. OSD is becoming increasingly common in industry. Vendors such as IBM Rational are promoting their Jazz technology platform (www.jazz.net) as an extensible team collaboration platform aimed at integrating people, process and assets across software development projects. University researchers in the U.S.A, Canada and Germany have received Jazz grants in 2008 to assist them with preparing student teams' skills in managing projects which are distributed across geographic and institutional boundaries (www.thetartan.org/2008/3/24/news/software). CollabNet (www.collab.net) is a widely used collaborative platform (www.tigris.org) that supports globally distributed software development teams. Recently, Chair of SE, Prof Meyer at ETH, Zurich has started offering distributed and outsourced SE projects for student teams over several universities to collaborate (<http://se.ethz.ch/teaching/2008-H/dose>).

In Australia, you must graduate from an accredited engineering program to be assured graduate membership of Engineers Australia (EA). Assessment for accreditation by EA is based on the curriculum: structure and content, the teaching and learning environment and the quality assurance framework (<http://www.ieaust.org.au/membership/accreditation.html>).

Since 2002, BSE students at Monash University undertake a full year (2 semester) capstone project unit where they work in teams on a large software project for a client. The clients are from the industry or research organisation. The objective of the final year SE capstone project is to expose the students to real-life project scenarios and current (global) SE development practices. The teams of 4-5 Monash final year SE students are formed at the beginning of the year. They follow an agile process and work in pairs for incremental releases. The 4-5 member teams develop an architectural and design model and divide up their tasks taking dependency and subsystem interactions into consideration. They focus on program interface descriptions and learn the importance of such SE principles by doing. The architecture-centric component-based development is especially important in agile projects with offshore setting [2].

One of our industry clients for SE capstone projects has expanded his development practices in India and has wanted to institute the offshoring model for the project with Monash SE students in 2008. This is in line with the accrediting body, Engineers Australia's philosophy that educational institutions must "get closer to industry" as part of engagement with external constituencies, be aware of and be responsive to industry demands for quality SE graduates. Engineers Australia is particularly keen on our MUSE studio projects which assesses students' understanding of the key areas of software engineering body of knowledge.

In this paper, we report on our preliminary work on adapting our software engineering capstone projects to agile offshoring. We have migrated from our earlier customised work with a project management, code management and process-oriented paradigm to a more integrated open-source tool-based approach.

2 Variations from real-life industry offshore projects

In the SE capstone projects conducted by the final year student teams in an university, couple of issues such as costs/budget and time limits associated with the project does not reflect reality. In our MUSE capstone projects, student teams conduct the industry client's project for no payment. There is a fixed time limit of 2 semesters of study (late February - end of October) imposed based on the university calendar. A legal agreement is completed for each MUSE capstone project and simulates a commercial consultancy situation for the final year students. The client agrees to provide the project team with access to data and systems in order to facilitate development of the system which forms the basis for assessment for the students. The student assigns to Monash University ownership of all rights in any IP arising from the project including future copyright. The client owns the IP rights, including copyright, and grants the university a royalty free perpetual licence. A revised legal agreement must be used where there is an offshoring component for the project as the offshoring party may be a paid consultant or another student team from an overseas educational institution.

Also, capstone projects are expected to be deployed as working systems in the client's organisation. It is not appropriate for us to engage our students just in the early phases of a software development life cycle and ask the offshore partner to implement and test the applications. Such a distribution of work will not be beneficial to the learning outcomes envisaged for the final year full-year project. Our pedagogical approach requires student teams to learn by doing the various key areas of Software Engineering Body of Knowledge (SWEBOK) as part of their project.

3. Process-oriented capstone projects

Monash University Software Engineering (MUSE) Studio lab is equipped to access commercial tools with academic licences and open source tools in SE development environments, methods, testing and configuration management. Student teams are required to use these SE tools and follow agile processes such as pair programming, collaboration, teamwork in SE process and product development, and testing, and for continuous integration and incremental software releases. The studio is used for practical applied SE tasks and for preparing students as professional software engineers in the IT industry locally or as contractors in the globalised workforce.

In 2002-2003, our emphasis was on using a tailored middle-weight process which requires students to devote enough effort in requirements and design models and move

more to the agile method during implementation and testing. However, their SE product and process artifacts were not accessible uniformly in a virtual environment. In 2004, a service-oriented portal was introduced for the capstone project teams [3]. It acted as a front end for exposing services such as task tracking and project planning for these teams. Support infrastructure services of user authentication, role and service allocation for members etc were managed through the portal environment.

Prior to this Portal service, students kept track of their project tasks as there was no online central repository set up for the teams. Also, since the lecturer was not able to track the teams' tasks online, there was a time lag in managing this process. The task tracker service provided by the Portal streamlined and improved data collection, reporting and sharing of project life cycle information. This ability to monitor and share detailed project status information enabled student teams to manage their project resources more effectively. For example, they were able to monitor tasks on-line, see which are slipping from the schedule, and thus allocate appropriate resources to ensure its completion. Since it is an e-service, the lecturer was able to keep track of student teams on-line and offer appropriate advice as needed. Other services such as file management, CVS etc were also available on the Portal server as project team accessible assets. However, collaboration features such as issues, bug tracking and revision control were not available as integrated portal services. In 2006, we moved from using CVS version control system to SVN and started migrating from the custom-built Portal to the school's servers and repositories for storing project and process assets as the Faculty restructuring resulted in some consolidation of technical infrastructure support. We are currently working with more integrated open source collaborative development platforms that support distributed teams and are also exploring platforms and tools that foster collaborative software development such as IBM Rational's Jazz technology platform and CollabNet tools to support our requirement to support OSD in capstone SE projects for Monash BSE program.

References

- [1] A. Kornstadt and J. Sauer. Mastering Dual-Shore Development - The Tools and Materials Approach Adapted to Agile Offshoring. In *SEAFOOD 2007, LNCS 4716*, pages 83-95. Springer-Verlag Berlin, 2007.
- [2] A. Kornstadt and J. Sauer. Tackling Offshore Communication Challenges with Agile Architecture-Centric Development. In *6th Working IEEE/IFIP Conference on Software Architecture WICSA2007, Mumbai, India*, pages 6-9, Jan 2007.
- [3] S. Ramakrishnan. A Service-Oriented Portal for Software Engineering Education. In *CATE2006, Lima, Peru.*, Oct 4-6 2006.