Putting Requirements and Quality at the Core of Global Service Delivery: Current Efforts and Future Plans at Pace University

Olly Gotel
ogotel@pace.edu

Christelle Scharff
cscharff@pace.edu

Pace University
Ivan G. Seidenberg School of Computer Science and Information Systems
New York, NY, USA

ABSTRACT

This paper outlines some initial steps we have taken at Pace University to prepare Computer Science students for working in a service-oriented business environment. In such an environment, we suggest that software procurement and development become tasks that revolve around articulating needs and finding a way to match these needs to available supply, wherever this supply may be sourced. The capability to act as either a client or a provider in this context thereby demands the mastery of a number of skills. Firstly, the ability to determine the changing services that are required and to describe these needs so as to support on-demand service acquisition. Secondly, the ability to develop, assure and assess the quality of component services as part of a larger global supply chain. Our Software Engineering teaching emphasizes the centrality of requirements engineering and quality processes, and designs student projects that provide experience in co-production via distributed global software development. We believe these skills are fundamental to supporting service determination and delivery.

INTRODUCTION

Over the past few years, the way in which we procure and develop software has been changing to leverage the economic advantage of global markets. To prepare students for this way of working, we have been making incremental changes to the Computer Science curriculum at Pace University, both at the undergraduate and graduate levels.

Our position has been that there are some prerequisite and enduring skills that need to be emphasized to fully realize the benefits of service-orientation and so enable on-demand business. These skills revolve around determining what customers need and specifying these needs in a service-oriented manner, along with the capacity to develop quality software to implement such services, assured to a sufficient level of quality and dependability. These skills pertain to the disciplines of requirements engineering, quality assurance and software testing. These are core skills that we advocate and promote at the graduate level in the curriculum of a specialist Masters degree in Software Design and Development. Within this degree we offer a sequence of courses on Systems Requirements Engineering, Software Reliability and Quality Assurance, and Software Testing.

At the undergraduate level, we bring the essence of this graduate teaching into the capstone course on Software Engineering. This is a practical course that is centered on group project work. To give students requisite exposure to global working, we craft projects that enable students to experience multiple sides of the offshore outsourcing relationship. We have been working closely with the Institute of Technology of Cambodia and the University of Delhi to set up an innovative three-way partnership to achieve these objectives.

This paper outlines our efforts to date with the graduate courses and the integration of this knowledge into the project-based undergraduate

teaching. It highlights both why and how we propose these efforts support the move towards service orientation. We further illustrate how we have been using IBM technology in these courses, leveraging the benefits of the IBM Academic Initiative, and also suggest how we hope to capitalize upon and contribute to the proposed Shared Software Infrastructure (SSI) initiative to further support our efforts.

CENTRALITY OF REQUIREMENTS

The term 'service' has been defined as "a provider/client interaction that creates and captures value"[1]. 'Value' can be considered the quality that renders something desirable or worthwhile. Central to this definition is therefore an understanding of the stakeholders involved in an exchange and their respective needs. The area of Software Engineering that deals with these concerns is Requirements Engineering.

Our teaching in the area of Requirements Engineering focuses on stakeholder identification, needs elicitation and negotiation, and requirements description. Notably, the issue of who is a provider and who is a client is not always trivial; there is often a reciprocal element to consider if the engagement is to be mutually beneficial. Students therefore learn about the dependencies and social networks that underpin any complex software development project. In addition, it is rarely sufficient to write a monolithic requirement specification, so students learn the skills required to break down needs into interlocking and smaller component areas, within a framework that is supportive of change and integration. These skills are critical to the articulation and definition of services.

How the satisfaction of a requirement is sourced is increasingly immaterial. It may be outsourced to a software house, capability may be purchased off the shelf or a service may be requested dynamically over the Internet. Writing requirements in terms of services, and conversely the task of describing software components in terms of services, is a focal aspect of our requirements training.

ASSURING AND ASSESSING QUALITY

In a service-oriented world, one can select off the shelf services, on the fly, to satisfy dynamically changing requirements. The issue here becomes one of not only ensuring that the selected service does what is required, but also one of trust or dependability. The service will have to satisfy specified functional requirements within certain constraints, be these with respect to performance, reliability or security, etc. There may also be the need to check that the selected service does not do anything that is not required, particularly when integrated into a wider systems context. Students need to acquire the skills that are essential to both assess candidate services and to assure services when initially developed prior to deployment. We provide training in these fundamental skills through a series of courses on Software Quality Assurance and Software Testing.

Our Software Quality Assurance course focuses on the critical link between requirements and software quality, and considers the issue of testing at a systems and user acceptance level. Our Software Testing course covers developer-level testing, such as unit testing. All our testing-related courses revolve around test planning, test design and development, test execution, test reporting and test exit criteria. Inspired by the practices of the Agile Methodologies, our teaching emphasizes test-first, test-driven development and the close involvement of the client when considering software quality factors. The general ability to assess and assure quality of software is a skill critical to deploying and procuring dependable services.

GLOBAL CO-PRODUCTION EXPERIENCE

Since 2005, the focus of our undergraduate capstone Software Engineering course has been global software development. In 2005, teams of Pace University and Institute of Technology of Cambodia students worked together to develop software products for the Cambodian market [1]. We organized the student projects so that: (a) the Cambodian students acted as clients and end-

[1] http://www.research.ibm.com/ssme/services.shtml.

users – they knew the problem the proposed system was to tackle, the environment it was to operate in and had the authority to accept the work of the providers (or not); and (b) the Pace University students acted as providers – it was their responsibility to 'capture' the requirements for the system, propose design options, develop the selected design and test the eventual system, while also handling requirements changes.

In 2006, this model was extended to include students from the University of Delhi. These students had considerable technical expertise in database design, so the concept of a lead contractor and third-party supplier was introduced. While the Cambodian students remained as clients, the provisioning of the solution was changed. The Pace University students subcontracted part of the system design and development to the students from India, while also managing the end-to-end contract.

The intention behind this effort was to provide students with a realistic co-production experience where software products get engineered by global partners with disparate skills and expertise. The US and Indian students obviously had to work closely together to develop a product for a Cambodian client. Such a project requires students to learn about the delineation of responsibilities and the management of changing expectations across supply chains. One of the important skills they must develop is learning how to elicit requirements from a distance and how to write these requirements in such a manner so as to validate understanding, support change, and facilitate distributed working and continuous integration. The Software Engineering course thereby becomes a wider learning experience where students find out about the processes and measurements involved in setting up, running and evaluating a service engagement. They discover the need for contracts and agreements, relationship management and softer communication skills. More significantly, they learn about different cultures, professional regulations and motivations, and develop skills to address business and technical issues in a service-based business environment.

SUPPORTING TOOLS

Our graduate courses introduce students to some of the IBM product offerings in the requirements and testing areas. Specifically, Rational Requisite Pro is used to illustrate the importance of requirements description, requirements management and requirements validation. Our Software Testing courses introduce students to IBM Functional Tester to test GUI applications. Here we focus on the capabilities of automated functional testing and regression testing via Java scripting. Our courses also emphasize open source and open standards, with Eclipse/JUnit typically being used for unit testing.

The undergraduate capstone Software Engineering course introduces students to some of the IBM product offerings and open source technology supporting the Software Engineering process. Students are introduced to the Unified Modeling Language (UML) and use IBM Rational Modeler as a stand-alone application to model design options for realizing the software specification, thereby achieving a better understanding of how the system should behave and correspond to client needs. This also facilitates the communication between designers in India and developers in the US. The developed software is generally web-based Java applications with back-end databases implemented in MySQL. The Eclipse development environment is regularly used so that developers can take advantage of the wider support tools. Students use the JUnit plug-in for unit testing and CVS for code sharing, change and version management, as well as the externalization mechanism to internationalize the software -- the software needs to be delivered in English and French for a Cambodian audience. Trac[2] is an open source wiki-based software solution for supply chain and project management. Clients use trac to report bugs and developers use trac to fix and manage bugs.

[2] http://www.edgewall.com/trac/.

FUTURE PLANS

We have placed an emphasis on process issues over the past two years in order to determine the underlying skills that need to be established for students to move towards implementing and using service-enabling technology. Our strategy has been to introduce these core supporting concepts and technologies incrementally. We consistently rely on open source development environments and standards in our courses. Through the IBM Academic Initiative, we have been using IBM technology in our courses to support the different phases of the Software Engineering process. One of the limitations we have found has been the fact that the current offering for students includes only IBM Rational Modeler, IBM Rational Developer, IBM Application Websphere and IBM DB2 Personal Developer. This constrains students to use evaluation versions of IBM Rational Requisite Pro and IBM Rational Functional Tester.

One issue for smaller schools is the obvious cost incurred in installing, configuring and managing an end-to-end technical infrastructure for students to specify, design and build their software products within. To this end, Pace University has been participating in an initiative coordinated by IBM to promote a Shared Service Infrastructure (SSI) between schools. Over the next academic year, we hope to participate further in this initiative by determining the software and configuration requirements needed to support core courses such as ours, particularly as we build upon the enabling skills and move towards service-orientation. We view this as a crucial step towards overcoming some of the barriers to technology use in the classroom. Particular attention will be given to the configuration requirements to enable global distributed software development. The intention of the SSI initiative is also to share courseware. We hope to contribute experiences from our courses, in particular the studies from the global supply chain projects, to help put requirements and quality concerns at the heart of global service delivery training.

Our current challenge as Computer Science instructors is to provide students with opportunities and situations in which they can experience the multi-disciplinary nature of the computing field, integrating training in skills from traditional engineering, the social sciences and management. This implies the development of courses or Computer Science curricula that focus more on services and the alignment of business needs to technology – not only on teaching technology in isolation from real-world applications and working contexts. We have begun to respond to these issues at the graduate and undergraduate levels, in particular with the global co-production experiment. We plan to collaborate further with IBM and the New York software industry to import (and then export) service-centered software development practices into our project-based courses.

Agile methodologies represent an advance in software processes, focusing on integrating people management, process and technology. They promote close working relationships with customers and end users to elicit and check requirements through a continual dialogue. As researchers, we are particularly interested in distributed agile software development if we are to fully realize the benefits of dynamic service provisioning.

Through our collaboration with IBM, we organized a seminar series for students and faculty in fall 2005 and spring 2006. In these seminars, students and faculty were introduced to the latest IBM strategies, research and technologies. The 2005 series paid particular attention to service-oriented architecture and web services as a precursor to the establishment of a graduate course on this topic. We believe that collaborations such as these are a good way to initiate new ideas and encourage curriculum changes that align with industrial needs.

REFERENCES

1. Gotel O., Scharff S. and Seng. S. Preparing Computer Science Students for Global Software Development. *36th ASEE/IEEE Frontiers in Education Conference*, 2006 (to appear).