Chapter IX
A Framework for Success in Real Projects for Real Clients Courses

David Klappholz
Stevens Institute of Technology, USA

Vicki L. Almstrum
The University of Texas at Austin, USA

Ken Modesit
Indiana University – Purdue University Ft. Wayne, USA

Cherry Owen
The University of Texas of the Permian Basin, USA

Allen Johnson
Huston-Tillotson University, USA

Steven J. Condly
HSA Learning & Performance Solutions, USA

ABSTRACT

In this chapter, we demonstrate the importance of Real Projects for Real Clients Courses (RPRCCs) in computing curricula. Based on our collective experience, we offer advice for setting up an effective support infrastructure for such courses. We discuss where and how to find clients, the types of projects that we have used, and how to form and train teams. We investigate the variety of standards and work products that we have used in our courses and explore issues related to assessment and evaluation. Finally, we consider the benefits of an RPRCC-centric approach to computing curricula.

A course is underway. Students are excited, engaged, eager to apply what they are learning, eager to communicate with one another about their project work, what they need to accomplish, and what they must find out from outside stakeholders. As a lovely bonus, the project the students are developing is more than a toy problem or a product that will gather dust on the back of the shelf — they are writing software that is useful and will be used.

This type of course exists and has been successful in many settings, including public and private institutions, small, medium, and large institutions, and Historically Black and Hispanic-Serving institutions (that is, the colleges and universities at which the co-authors teach). In this chapter, we promote the idea of Real Projects for Real Clients Courses (RPRCCs) and discuss key issues related to successfully planning for and executing them in a variety of settings.

INTRODUCTION

RPRCCs are courses in which students work in teams to develop real software for real clients, including faculty and staff from their own institutions, for-profit companies, not-for-profit organizations, and government agencies. To be "real," software must meet the needs of the client by solving a problem or providing a service for the client or the organization the client represents. RPRCCs are appropriate in all Computing Curricula 2005 (Joint IEEE CS/ACM Task Force, 2005) disciplines, that is, computer science (CS), information systems (IS), computer engineering (CE), software engineering (SE), and information technology (IT), which we refer to collectively as "computing disciplines" or simply as "computing." RPRCCs are also appropriate in the full range of post-secondary institutions, including community colleges, four-year colleges, and universities, and can even be used at the secondary level.

This chapter explores the core issues covered in a taxonomy that has been developed by the co-authors over a number of years. The taxonomy, which delineates issues involved in designing and delivering RPRCCs, has been refined using feedback from participants in workshops and other conference activities (e.g., Almstrum, Klappholz, & Modesitt, 2007; Klappholz, Almstrum, & Modesitt, 2006). Appendix A gives the top two levels of the current version of the taxonomy.

In this chapter, we explore the following basic issues involved in developing and teaching an RPRCC:

- Client-related issues, including where to find them, how to vet them for appropriateness as clients, and how to manage client expectations;
- Project-related issues, including possible types of projects and how to vet projects for appropriateness;
- Team-related issues, including how to form teams and train them;
- Product-related issues, including standards and required work products; and
- Issues related to assessment and evaluation.

The full taxonomy details these and a large number of additional issues. Finally, in the Future Trends section, we argue for the notion of RPRCC-centric computing curricula, that is, curricula that include RPRCCs at multiple levels of the undergraduate program.

The experiences we discuss in this chapter can help readers understand the issues one must consider when planning the framework for an RPRCC. We sincerely hope that the ideas presented below will better equip instructors with all types of experience to plan and execute successful RPRCCs.

Related Content

The Effects of the State of Union Address of USA Presidents to Their Performance Ratings
Carl Lee (2015). *International Journal of Software Innovation (pp. 1-12).*
www.irma-international.org/article/the-effects-of-the-state-of-union-address-of-usa-presidents-to-their-performance-ratings/122788

VirtualCareGiver: Personalized Smart Elderly Care
Seiki Tokunaga, Kazunari Tamamizu, Sachio Saiki, Masahide Nakamuraand Kiyoshi Yasuda (2017).
International Journal of Software Innovation (pp. 30-43).
www.irma-international.org/article/virtualcaregiver/169916

Integrating Semantic Web and Software Agents: Exchanging RIF and BDI Rules
Yiwei Gong, Sietse Overbeekand Marijn Janssen (2011). *International Journal of Systems and Service-Oriented Engineering (pp. 60-76).*
www.irma-international.org/article/integrating-semantic-web-software-agents/55062

An Overview of Software Quality Concepts and Management Issues
Alain Apriland Claude Y. Laporte (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 222-241).*
www.irma-international.org/chapter/overview-software-quality-concepts-management/29391

Software Modeling and Design Process
Smrity Dwivedi (2023). *The Software Principles of Design for Data Modeling (pp. 65-75).*
www.irma-international.org/chapter/software-modeling-and-design-process/330487