Chapter II

Augmented WebHelix: A Practical Process for Web Engineering

Nary Subramanian
University of Texas at Tyler, USA

George Whitson
University of Texas at Tyler, USA

ABSTRACT

Process is an important element in the success of any information systems development project, especially in academia where typically an undergraduate term project needs to go through the development phases within the space of a semester. Traditionally academic processes have been adapted versions of well-known industrial processes with one major exception—lack of customer feedback in the process. This omission of customer feedback results in students completing “toy” projects without significant real-world experience; efforts to incorporate artificial customer interactions have not been very successful either. It is our opinion that the industry processes cannot be simply copied in academia; what is required is a process that will better equip the students to face real-world challenges. WebHelix has been recently introduced as a practical process for Web engineering that helps students gain valuable real-world experience without sacrificing project and product management phases. In this chapter we propose the Augmented WebHelix process that augments the WebHelix in three ways: provides an option at the end of each slice of the helix to both release the current version and continue to the next slice of development; provides a qualitative evaluation framework, called the project evaluation framework (PEF), that provides a systematic approach for evaluating the status of the project; and the ability to evaluate the project at the end of each phase in a slice of the helix. The first augmentation provides the ability to release and continue which is more practical than the go/no-go approach adopted by WebHelix; the second augmentation, the PEF, allows different factors besides the return-on-investment as in WebHelix to be considered for evaluating the current phase and status of the project, and the third augmentation provides the ability to ensure the project is on track. In this chapter we describe the augmented WebHelix process and demonstrate its applicability to both academia and industry with examples.
1. INTRODUCTION

A Web application is an information system that can deliver complex content and functionality to a broad population of end users [17] and consists of a set of Web pages that are generated in response to user requests [5]. Examples of Web applications include search engines, online stores, auctions, news sites, instructional sites, and games. Features of a Web application include substantial published content, a complex navigational model, a complex data design, many computational modules, and security considerations.

Most companies have a Web presence today. The initial Web presence for a company is often an informational Web site with little or no database use. Because of the simplicity of informational Web sites, companies often underestimate the difficulty of upgrading such a site to a database-driven Web application. They see little need to use a formal software engineering process and this generally produces a Web application that is behind schedule, not fully functional and almost impossible to upgrade. Experienced project managers realize that developing a Web application could be a complex effort and that some form of systems analysis and design is needed. But, projects usually have small budgets and short deadlines, so if too much time is spent in the design phase, projects usually do not get finished on time (this has been referred to as the analysis-paralysis problem in [26]).

A similar situation exists in academia as well. The majority of the courses in computer science (CS) and computer information systems (CIS) programs require term projects to be completed during the semester. Very often the upper-level courses require the term projects to be team projects as well with the expectation that the students will familiarize themselves with the real-world system and software development processes. All the projects, almost without exception, go through the broad phases of analysis, design, and implementation. Many of the courses stress to a certain degree the concepts such as software process, divide-and-conquer, establishment of phases, and project management, with the belief that some understanding of these core ideas will help the students complete the semester project while at the same time, in many cases, doing course work besides taking the full load of classes for the semester. Our experience in the past has been that very frequently students do not complete or in many cases skip altogether the important phases of analysis and design for the term projects defeating the very purpose of preparing the students for the industry.

Most of the courses focus on using one of the software development processes such as the waterfall, incremental, object-oriented, extreme programming (XP), rational unified process (RUP), or system development life cycle (SDLC) [17, 26, 27]. Our analysis for the reasons for many of the problems faced by the students in their project courses has led us to the conclusion that simply adapting an industry-strength process does not help in academia. One of the most important reasons seems to be that the current processes used in the industry and recommended for use in academia, suffer from one big requirement - the feedback loops require actual customer participation. Most academic projects do not have realistic customer feedback resulting in the following problems:

1. **Students imagine the feedback:** Students go through the motions of updating their artifacts based on imagined customer feedback; while role-playing and reviews have their uses in software development, their primary purpose is not to replace the customer and this imagination approach tends to make the feedback mechanism almost unimportant;

2. **Students ignore feedback:** The assumption is that “let us assume that the customer has approved the artifacts of the previous phase”—this assumption is very closely related to imagined feedback; the realistic
Related Content

Autonomic Healing for Service Specific Overlay Networks
[www.igi-global.com/article/autonomic-healing-service-specific-overlay/70385?camid=4v1a](www.igi-global.com/article/autonomic-healing-service-specific-overlay/70385?camid=4v1a)

A Scalable QoS-Aware Web Services Management Architecture (QoSMA)
[www.igi-global.com/chapter/scalable-qos-aware-web-services/5222?camid=4v1a](www.igi-global.com/chapter/scalable-qos-aware-web-services/5222?camid=4v1a)

Performance Analysis of a Web Server
[www.igi-global.com/article/performance-analysis-web-server/2653?camid=4v1a](www.igi-global.com/article/performance-analysis-web-server/2653?camid=4v1a)

All-Optical Internet: Next-Generation Network Infrastructure for E-Service Applications
[www.igi-global.com/chapter/all-optical-internet/5225?camid=4v1a](www.igi-global.com/chapter/all-optical-internet/5225?camid=4v1a)