Chapter XI

Evolution and Maintenance of Web Sites: A Product Line Model

Rafael Capilla*, Universidad Rey Juan Carlos, Spain

Juan C. Dueñas**, Universidad Politécnica de Madrid, Spain

ABSTRACT

In this chapter we describe the product line models, and show how to apply them for developing and evolving Web products. A product line captures the common and variable aspects of software systems as key assets under a common architecture. Software companies are increasingly adopting this approach in order to accelerate the development of families of similar software products. In certain domains, such as the Web systems, development and maintenance operations are required more often. New techniques to engineer Web sites are needed in order to reduce the time to market for the Web products and to maintain the systems afterward. The authors believe that understanding the notion of lightweight product line and the role that the architecture plays will help software engineers in the construction of software products, and they will be able to manage the evolution effectively against future changes.

INTRODUCTION

For several years, software engineers have been taught to build software systems from a set of technical requirements. Requirements lead to a software design and finally to a software system. As far as we know from our experience, software systems must be continuously improved or fixed due to new requirements or problems not previously identified. The notion of software evolution (Lehman, 2001) aims at the improvement of
existing systems in order to correct faults, to adapt them to new changes or to develop new versions, among others. Nevertheless, software evolution is not an isolated concept. In fact, the evolution of a system is closely related to the software maintenance process. Although some authors consider them as synonyms, we understand that the proper evolution of a system relies on specific software maintenance operations. Software evolution generally describes the changes a system suffers in the long term, regarding unexpected sources for change (Kniesel, Constanza & Dmitriev, 2002). In contrast, the term software maintenance is usually employed for describing a set of activities or tasks for correcting, adapting or improving a software system.

There is a general consensus that maintenance processes are time and cost consuming. In order to reduce this impact, we need to employ appropriate techniques to reduce the time spent in maintenance processes as well as the budget needed to support the evolution of a system. The maintenance activity is considered as an integrated phase under the development process, and the use of specific methods and models for maintenance operations is frequently forgotten. This situation usually arises in those applications with strong time to market requirements and short product delivery deadlines, so the time to perform changes or improvements in the code or even to plan and design changes is much less compared to the actual time required to do these effectively and properly. For example, Web-based applications allow very short development time, and maintenance operations for these systems are usually performed in days or weeks. In order to address these, both the development and the maintenance of Web systems (i.e., Web products) require agile methods in order to support the fast evolution of such systems. The main goal of this chapter is to study agile and efficient methods, such as lightweight product line architectural (PLA) models, and analyze the benefits expected in the development of software systems, employing successful techniques to improve the maintenance operations of Web products.

BACKGROUND OF PRODUCT LINE MODELS

A Product Line Architecture (PLA) defines a set of similar systems that share a common software architecture (Bass, Clements & Kazman, 2003), a group of features and satisfy the needs of a particular market segment. More formally, a software product line is “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” (Clements & Northrop, 2003). A key aspect in this definition is that a product line refers to a set of related systems as those ones belonging to a particular domain or market segment. In this way, product lines are strongly market-driven because they try to cover the needs of a market niche. Such products are usually similar and they share common elements. Other authors call them “product family”; in this chapter we use the terms “product family” and “product line” interchangeably.

Software architecture is very much related to the notion of product line model. A software architecture represents the design of a set of similar systems and is formed by components and connectors. The components in the architecture describe the functionality of parts of the system or sub-systems, whereas the connectors describe the way in
Related Content

A Post-Implementation Case Study and Review of Enterprise Resource Planning (ERP) Implementations
www.igi-global.com/chapter/post-implementation-case-study-review/44139?camid=4v1a

Technology Institutionalisation through Technological, Organisational, and Environmental Isomorphism
Azadeh Pishdad and Abrar Haider (2015). Business Technologies in Contemporary Organizations: Adoption, Assimilation, and Institutionalization (pp. 54-74).
www.igi-global.com/chapter/technology-institutionalisation-through-technological-organisational-and-environmental-isomorphism/120751?camid=4v1a
ICT and Web 2.0 Technologies as a Determinant of Business Performance
www.igi-global.com/chapter/ict-web-technologies-determinant-business/69115?camid=4v1a

Business Networking: The Technological Infrastructure Support
www.igi-global.com/chapter/business-networking-technological-infrastructure-support/28655?camid=4v1a