Supporting Incremental Product Development using Multiple Product Line Architecture

Bedir Tekinerdogan, Information Technology, Wageningen University, The Netherlands
Özgü Özköse Erdoğan, Aselsan, Ankara, Turkey
Onur Aktuğ, Aselsan, Ankara, Turkey

ABSTRACT

Software product line engineering (SPLE) has been successfully applied in various application domains to support systematic reuse. Besides of its benefits it is also acknowledged that the SPLE process can in practice be considered too time consuming and heavyweight due to the required planning and development of the asset base. For this reason more lightweight SPLE processes are required that can be integrated in the ongoing product development of the organization. In this context, the authors share their experiences in adopting a multiple product line architecture to support the incremental product development of Aselsan REHİS, a leading high technology company in Turkey. The authors first discuss the important business needs for defining a more lightweight multiple product line engineering (MPLE) process. Then they discuss the multiple product line architecture and how it has been used to guide the incremental product development.

Keywords: Incremental Process, Industrial Case, Introduction, Multiple Product Line Engineering, Software architecture design

1. INTRODUCTION

Software product line engineering (SPLE) aims to provide pro-active, pre-planned reuse at a large granularity (domain and product level) to develop applications from a core asset base. The benefits for adopting a product line approach have been analyzed and discussed before (Schmid & Verlage, 2002; Birk, 2003). By investing upfront in preparing the reusable assets, it is expected to develop products with lower cost, get them to the market faster and produce with higher quality. In alignment with these goals different software product line engineering processes have been proposed such as, the SEI’s Framework for Software Product Line Practice (Clements & Northrop, 2002), the Fraunhofer’s PULSE-approach (Bayer, 1999), the Philips’ CoPAM method (America et al., 2000), the FAST approach (Weiss & Lai, 1999), and the Gomaa’s PLUS approach (Gomaa, 2005). Although different processes

DOI: 10.4018/ijkss.2014100101
have been proposed they share the same concepts of domain engineering, in which a reusable platform and product line architecture is developed, and application engineering, in which the results of the domain engineering process are used to develop the product members. Studies have shown that the SPLE process has been successfully applied in different application domains. On the other hand, for several application domains and organizations the demanded long-term investment and effort is technically and economically less feasible. The changing and strict market conditions have led to the need to customize the SPLE process to meet the stringent time-to-market constraints and be more flexible in handling changes. Incremental software development methods have been introduced for a long time and are also frequently applied in practice to promote adaptive, responsive approach to requirements changes. Initially, both paradigms, that is, SPLE and incremental development, have evolved separately. However, recently, several studies have proposed the application of lightweight approaches to SPLE to benefit from both paradigms (Carbon, 2006). The aim here is to support reuse-driven development while at the same time adopting a lightweight process that can embrace changes. Although theoretical comparisons have been provided, still more experiences from industrial practices are needed to validate the hypothesis raised by these studies. In this paper we indeed aim to share our own experiences in realizing an incremental and iterative product line engineering process within the industrial context of Aselsan REHIS, a leading high technology company in defense systems development in Turkey. The paper presents the results of the last seven years multiple product line engineering (MPLE) project REFoRM for defense applications. The adoption of MPLE provides a further increase of the granularity of reuse but due to the larger scope of the MPL the upfront investment and the time-to-market of the first products will be substantially higher, which is not feasible. To cope with these issues, the need for adopting lightweight approaches becomes even more important in an MPLE setting. To guide the product development process we have used the MPL architecture. Based on the MPL architecture the overall product line engineering process has been customized and the overall strategy for product development has been defined. We discuss the important business needs of the Aselsan MPLE project, describe the adopted incremental process and our lessons learned within the context of Aselsan REHIS.

The remainder of the paper is organized as follows. In section 2 we discuss the background and the preliminaries for the subsequent sections. Section 3, describes the industrial case of Aselsan REHIS for setting up a multiple product line engineering process, together with the needs for customizing the product line engineering processes. Section 4 describes the adapted MPLE process and its application in the industrial context. In section 5 we provide the related work and finally section 6 concludes the paper.

2. PRODUCT LINE ENGINEERING PROCESS

Based on the literature study on the existing SPLE processes we can derive a common SPLE process. In general there appears to be a consensus that the SPLE process consists of life cycle processes of domain engineering and application engineering. The domain engineering process is responsible for establishing the reusable platform and thus for defining the commonality and the variability of the product line (Pohl, 2005). The platform consists of all types of software artefacts (requirements, design, realization, tests, etc.). In the application engineering process the applications of the product line are built by reusing the artefacts and exploiting the product line variability as defined in the domain engineering process. Each of the sub-processes in the domain engineering and the application engineering processes result in so-called development artefact (Pohl, 2005). The development artefacts developed in the domain engineering process are called...
Related Content

Optimizing User Quality of Experience through Overlay Routing, Bandwidth Management and Dynamic Trans-Coding

[www.igi-global.com/chapter/optimizing-user-quality-experience-through/63580?camid=4v1a](www.igi-global.com/chapter/optimizing-user-quality-experience-through/63580?camid=4v1a)
EAST-ADL: An Architecture Description Language for Automotive Software-intensive Systems in the Light of Recent use and Research
[www.igi-global.com/article/east-adl/156512?camid=4v1a](www.igi-global.com/article/east-adl/156512?camid=4v1a)

A Novel Framework for Efficient Extraction of Meaningful Key Frames from Surveillance Video
[www.igi-global.com/article/a-novel-framework-for-efficient-extraction-of-meaningful-key-frames-from-surveillance-video/129492?camid=4v1a](www.igi-global.com/article/a-novel-framework-for-efficient-extraction-of-meaningful-key-frames-from-surveillance-video/129492?camid=4v1a)

Generating Indicators for Diagnosis of Fault Levels by Integrating Information from Two or More Sensors
[www.igi-global.com/chapter/generating-indicators-diagnosis-fault-levels/69673?camid=4v1a](www.igi-global.com/chapter/generating-indicators-diagnosis-fault-levels/69673?camid=4v1a)