Dealing with Cost Estimation in Software Product Lines: Experiences and Future Directions

Andy J. Nolan$^1$ and Silvia Abrahão$^2$

$^1$Rolls-Royce
SIN C-3, Rolls-Royce plc, PO Box 31
Derby DE24 8BJ, England
Andy.Nolan@Rolls-Royce.com

$^2$ISSI Research Group, Department of Computer Science
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022, Valencia, Spain
sabrahao@dsic.upv.es

Abstract. After 5 years invested in developing accurate cost estimation tools, Rolls-Royce has learnt about the larger potential of the tools to shape many aspects of the business. A good estimation tool is a “model” of a project and is usually used to estimate cost and schedule, but it can also estimate and validate risks and opportunities. Estimation tools have unified engineering, project and business needs. The presence of good estimation tools has driven higher performance and stability in the business. It was evident we needed this capability to underpin decisions in our new Software Product Line strategy. The objective of this paper is twofold. First, we report the experiences gained in the past on the use of estimation tools. Second, we describe the current efforts and future directions on the development of an estimation tool for Software Product Lines. At the heart of the Product Line estimation tool is a simple representation of the product – represented as the number of Lines Of Code (LOC). The next generation of tool, will need to consider wider aspects of product quality in order to create more accurate estimates and support better decisions about our products.

Keywords: Cost Estimation, Software Product Lines, Industrial Experiences.

1 Introduction

The production of quality software, on time, and within budget, remains an open problem of Software Engineering that has been addressed from different approaches. An industrial approach to this problem is to use Software Product Lines (SPL). Several benefits are associated to the introduction of product lines in software development organizations such as cost reduction, time-to-market improvement, project risk reduction, and quality improvement.

However, the associated costs and the quality of the software products may greatly differ due to systematic reuse. In addition, product line engineering is often the more economical choice in the long-term run. It might not be the best choice when project managers want to amortize their core asset base across only a few products or across
products with little commonality [1]. Therefore, there is a need for tools to help project managers to analyze in which situations and scenarios product line investment pays. To address this issue, several cost estimation models for Software Product Lines (SPL) have recently been proposed in the literature. However, to understand their benefits and weaknesses, it is important to analyze the experiences gathered in applying these models in industrial or organizational settings.

In this paper, we present an experience report about the use of cost estimation tools at Rolls-Royce. The objective of this paper is (i) to report the experiences gained in the past on the use of a cost estimation tool based on COCOMO (Constructive Cost Model) [3] (ii) to describe how this tool was extended for its use with software product lines as well as the lessons learned (iii) to describe future extensions for this tool based on the preliminary results obtained within the MULTIPLE (Multimodeling Approach for Quality-Aware Software Product Lines) project conducted at the Universidad Politécnica de Valencia in Spain with close collaboration of Rolls-Royce.

This paper is organized as follows. Section 2 discusses existing models and tools for cost estimation in SPL. Section 3 discusses past experiences on the use of a Cost Estimation tool at Rolls-Royce. Section 4 presents an overview of the SPL initiative launched in 2008 as well as the development of an estimation tool which was built for assessing the benefits of SPL. Section 5 describes the lessons learned and the future extensions of the tool. Section 6 presents our conclusions and further work.

2 Related Works

In the last few years several cost estimation models for software product lines have been proposed. Some representative proposals are: [16], [1], [19], [4], [7], [10], [9] and [13]. Poulin [16] proposed one of the first models for analyzing the effects of employing a systematic reuse approach. The model is based on two parameters: the relative cost of reuse (RCR) and the relative cost of writing for reuse (RCWR). The first parameter can be used for comparing the effort needed to reuse software without modification to the costs associated with developing the same software for a single use. The second parameter relates the costs of creating reusable software to the cost of creating one-time use software. These parameters can also be applied in the context of software product lines. The Poulin model uses the RCR and RCWR to calculate two other indicators (i.e., reuse cost avoidance and additional development cost) that predict savings for developing a specific project.

Böckle et al. [1] proposed a software product line cost model to calculate the costs and benefits that we can expect to have from various product line development situations. In particular seven reuse scenarios were identified. The cost model proposed involves the following four costs: (1) the cost to an organization of adopting the product line approach for its products; (2) the cost to develop a core asset base suited to support the product line being built; (3) the cost to develop unique software that is not based on a PL platform; (4) the cost to reuse core assets in a core asset base. The authors then analyze the cost savings of converting products to a software product line as they evolve over time.

Tomer et al. [19] proposed a model that enables software developers to systematically evaluate and compare alternative reuse scenarios. The model supports the clear