Computer Languages, Systems & Structures ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Evolutionary robust optimization for software product line scoping: An explorative study

Reza Karimpour *, Guenther Ruhe

Software Engineering Decision Support Lab, Computer Science Department, University of Calgary, Canada

ARTICLE INFO

Article history:
Received 2 December 2015
Received in revised form
12 July 2016
Accepted 13 July 2016

Keywords:
Search-based software engineering
Software product line scoping
Robust optimization
Evolutionary optimization

ABSTRACT

Background: Software product line (SPL) scoping is an important phase when planning for product line adoption. An SPL scope specifies: (1) the extent of the domain supported by the product line, (2) portfolio of products in the product line and (3) list of assets to be developed for reuse across the family of products.

Issue: SPL scope planning is usually based on estimates about the state of the market and the engineering capabilities of the development team. One challenge with these estimates is that there are inaccuracies due to uncertainty in the environment or accuracy of measurement. This may result in issues ranging from suboptimal plans to infeasible plans.

Objective: To address the above, we propose to include uncertainty as part of the SPL scoping model. Plans developed in consideration of uncertainty would be more robust against possible fluctuations in estimates.

Approach: In this paper, a method to incorporate uncertainty in scoping optimization and its application to generate robust solutions is proposed. We capture uncertainty as part of the formulation and model scoping optimization as a *multi-objective* problem with *profit* and *stability* as fitness functions. Profit stability and feasibility stability are considered to represent stability concerns.

Results: Results show that, compared to other scope optimization approaches, both performance stability and feasibility stability are improved while maintaining near optimal performance for profit objective. Also, generated results consist of solutions with trade-offs between profit and stability, providing the decision maker with enhanced decision support.

Conclusion: Multi-objective optimization with stability consideration for SPL scoping provides project managers with a robust and flexible way to address uncertainty in the process of SPL scoping.

1. Introduction

Software product line (SPL) is a development approach built on the idea that similar systems, which vary in some details, can be developed out of a single architecture. The benefits of this approach are observable in cost cuts and market agility [1]. Since the adoption of SPL requires significant upfront investment [2], it is plausible to validate the economic justification and profitability of such decision. Achieving this requires clear understanding of the domain and scope of product line.

* Corresponding author.
E-mail addresses: reza.karimpour@ucalgary.ca (R. Karimpour), ruhe@ucalgary.ca (G. Ruhe).

http://dx.doi.org/10.1016/j.cl.2016.07.007
1477-8424/

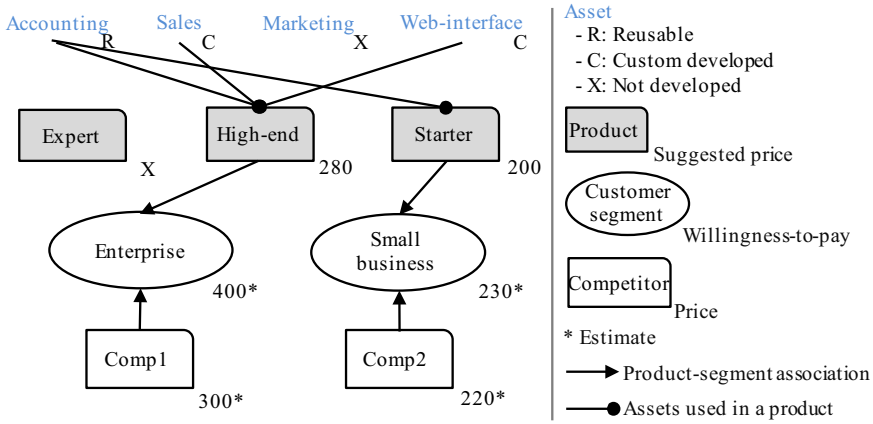

Fig. 1. A scoping example.

SPL Scoping is an activity concentrated on scope related decisions including Asset Scoping, Domain Scoping and Portfolio Scoping. Asset Scoping includes decisions about which assets of a product line are to be developed for reuse across different products. Portfolio Scoping includes decision about which products to be included as part of the product line portfolio. The final category of decisions in SPL Scoping is Domain Scoping that includes decisions about the extent of the domain supported by the product line [3].

An example of scoping is given in Fig. 1. This SPL consists of three candidate products, two customer segments and two competitors. As demonstrated, the product *High-end* is allocated to *Enterprise* segment and priced as 280. Starter product is allocated to the *Small business* segment and *Expert* product is not selected for production. As for Asset Scoping, *Accounting* feature is planned to be built for reuse while *Sales* and *Web-interface* are custom developed for each product. *Marketing* feature is skipped since it is exclusive to the *Expert* product. In summary, in this study, the decision of which product to build, what product to offer to which segment, and how to price a product comprise the scoping plan.

As reported in industrial case studies, among the three scoping types, Portfolio Scoping and Asset Scoping require more effort and resources [4]. Also, Portfolio Scoping and Asset Scoping greatly influence development activities. The two major development activities, *Asset development* and *Product development* [5], are affected by the scope defined in pertaining scoping activities.

Current methods that model SPL scoping are mainly based on return-on-investment (ROI) [6]. In these approaches, profit is the main directing measure and is defined as a function of cost and revenue: $profit = revenue - cost$ [7]. For example, Müller [7] model scoping as a profit maximization problem and Gillain et al. [8] developed goal based requirement modelling and NPV based valuation.

1.1. Motivation

1.1.1. Robust SPL scoping

Once adopted, an SPL may help reduce risks associated with software production [3,9]. Nevertheless, the decision of whether to produce a specific product or not and if a specific asset should be developed as reusable requires careful evaluation. One main reason for this is that most scoping approaches depend on some estimated input data. This input data includes cost estimates and market assessment to name but two. Depending on the type project, this estimation may come from stakeholders, marketing, development team, or management [10]. Most research papers do not address inaccuracies and uncertainties in estimates [10,11].

Many software engineering tasks rely on estimates that may be inaccurate, insufficient and are likely to change [12]. In the case of SPL scoping, estimated measures are, among others, *willingness-to-pay*, *production cost*, and *competitors*. Willingness-to-pay stands for the cost a customer is willing to pay for a product. On the other hand, production cost depicts the effort required to produce a product. Finally, estimates about competitors include their offerings and marketing strategies. Being estimates mean that these quantities may change as they are captured from environment and since the environment is not static, these numbers can also change over time. There is also the risk of error in the estimation process. For example, one may under-estimate the production cost of an asset while another may over-estimate willingness-to-pay of a customer segment.

1.1.2. Trade-off decision making

SPL managers need to make decisions early in the life cycle of a product line. Therefore, they need to consider the possibility of unexpected changes to a variety of variables. This often requires making trade-off decisions between choosing a very strict but fragile plan or deviate from optimality to gain robustness and stability.

Please cite this article as: Karimpour R, Ruhe G. Evolutionary robust optimization for software product line scoping: An explorative study. Computer Languages, Systems & Structures (2016), http://dx.doi.org/10.1016/j.cl.2016.07.007