Science of Computer Programming 78 (2013) 1010–1034

Contents lists available at SciVerse ScienceDirect

Science of Computer Programming

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

A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring

Miguel A. Laguna *, Yania Crespo

GIRO research group, University of Valladolid, Spain

ARTICLE INFO

Article history:
Received 11 April 2011
Received in revised form 10 May 2012
Accepted 14 May 2012
Available online 22 May 2012

Keywords:
Software product line
Evolution
Reengineering
Legacy system
Refactoring

ABSTRACT

Software product lines (SPLs) are used in industry to develop families of similar software systems. Legacy systems, either highly configurable or with a story of versions and local variations, are potential candidates for reconfiguration as SPLs using reengineering techniques. Existing SPLs can also be restructured using specific refactorings to improve their internal quality. Although many contributions (including industrial experiences) can be found in the literature, we lack a global vision covering the whole life cycle of an evolving product line. This study aims to survey existing research on the reengineering of legacy systems into SPLs and the refactoring of existing SPLs in order to identify proven approaches and pending challenges for future research in both subfields. We launched a systematic mapping study to find as much literature as possible, covering the diverse terms involved in the search string (restructuring, refactoring, reengineering, etc. always connected with SPLs) and filtering the papers using relevance criteria. The 74 papers selected were classified with respect to several dimensions: main focus, research and contribution type, academic or industrial validation if included, etc. We classified the research approaches and analyzed their feasibility for use in industry. The results of the study indicate that the initial works focused on the adaptation of generic reengineering processes to SPL extraction. Starting from that foundation, several trends have been detected in recent research: the integrated or guided reengineering of (typically object-oriented) legacy code and requirements; specific aspect-oriented or feature-oriented refactoring into SPLs, and more recently, refactoring for the evolution of existing product lines. A majority of papers include academic or industrial case studies, though only a few are based on quantitative data. The degree of maturity of both subfields is different: Industry examples for the reengineering of the legacy system subfield are abundant, although more evaluation research is needed to provide better evidence for adoption in industry. Product line evolution through refactoring is an emerging topic with some pending challenges. Although it has recently received some attention, the theoretical foundation is rather limited in this subfield and should be addressed in the near future. To sum up, the main contributions of this work are the classification of research approaches as well as the analysis of remaining challenges, open issues, and research opportunities.

1. Introduction

Software Product Lines (SPLs) are successfully used in industry to achieve a disciplined development of families of software systems [1]. SPLs combine systematic development and the reuse of coarse-grained components that include the

* Corresponding author.
E-mail addresses: mlaguna@infor.uva.es (M.A. Laguna), yania@infor.uva.es (Y. Crespo).

0167-6423/$ – see front matter
doi:10.1016/j.scico.2012.05.003

common and variable parts of the product line [2,3]. Many legacy systems, either highly configurable or with a long history of versions and local variations, are susceptible to reengineering as SPLs [4]. The catalog of techniques for evolving legacy software to become an SPL can be really extensive. Some are closely related to generic reengineering approaches. Others can be seen as model and code transformation techniques. In particular, refactoring techniques can be used to reconfigure the internal details of the legacy systems. However, the quality of an SPL resulting from the reengineering of legacy systems is often weak, especially regarding code and architecture quality, which is why refactoring techniques are often applied too. Existing SPLs can also be restructured using specific refactorings to improve their internal quality. Diverse academic proposals and industrial experiences addressing these topics are present in the literature, but a global vision that covers the whole life cycle of an evolving product line is missing. Some of our previous work was focused on SPLs [5,6] and refactoring [7,8], as separate topics. We were interested in the synergy of both fields to adapt existing techniques or to develop specific new ones to facilitate the conversion of (families of) legacy systems into an SPL, and/or evolve existing SPLs following a quality-driven process. With this aim, the need for a systematic search of the literature arises. This would help to clarify the envisioned possibilities and to establish a good foundation for our main goal, revealing remaining challenges, trends and open issues.

A systematic literature review uses a rigorous procedure for searching and analyzing previous research papers and has been adopted in Evidence-Based Software Engineering [9]. A systematic review tries to obtain all the relevant papers (the primary studies) for a specific question or area of interest, and to use them to summarize, assess and interpret the relevance of all evidence related to that question, minimizing possible bias. It is aimed at providing knowledge about when, how, and in what context technologies, processes, methods or tools are more appropriate for software engineering practices [10]. The phases of the review include planning (the research questions and the review protocol are defined), conduction (primary studies are identified, selected, and evaluated; for each selected study, data are extracted, analyzed and synthesized), and reporting (a final report is presented).

However, when a research area is evolving and there is an increasing number of results available, the research questions can be diverse and a previous effort of clarification and classification of the new approaches is required. To summarize, and to obtain an overview of the field of interest, mapping studies provide a comprehensive evaluation of research using a research strategy similar to systematic reviews [11]. A systematic mapping study allows the results that have been previously published to be categorized, and often gives a visual map of its results. Kitchenham et al. remark that mapping studies use the same basic methodology as systematic reviews but aim to identify and classify all research related to a broader software engineering topic, rather than answering questions about the relative merits of competing technologies [12]. They conclude that mapping studies provide a foundation to assist more focused research efforts.

We conducted our systematic mapping study in the software product line domain and aimed at identifying relevant primary studies related to the reengineering of legacy systems into SPLs or to the refactoring of these SPLs to improve their quality. It was initially conducted from October to December, 2010 and was carried out by two people (SPL and refactoring specialists). Second and third revisions were carried out at the end of 2011, and in April 2012, extending the search string with the "product family" related terms and incorporating 2011 studies. Adapting the suggestions proposed by Petersen et al. [11], the defined protocol included the following steps:

1. Definition of the research questions, the scope and strategy of the search, and the inclusion and exclusion criteria.
2. Conduction of the search for primary studies, using the established protocol and relevance and quality criteria (context, clear objectives, description of the research method) which enable the assessment of the quality of primary studies addressing bias and validity.
3. Screening of the papers, based on inclusion/exclusion criteria, and examining references to ensure the presence of as many articles as possible.
4. Classification of the papers, assigning diverse tags, following multifaceted criteria.
5. Data extraction and aggregation in several tables to have an overview of the field.

The rest of the paper is structured as follows. Some background on SPLs, refactoring and related terminology is introduced in Section 2. The following sections present how the above mentioned steps have been carried out in our systematic mapping. Section 3 describes how the mapping study methodology has been planned (step 1). Section 4 presents the details of the search and classification of the papers, compiling some statistics on the primary studies (steps 2, 3 and 4). Section 5 (legacy system reengineering) and 6 (product line refactoring) are organized into several blocks and tables and analyze the results of that classification (step 5). A discussion of the due and remaining challenges extracted from the mapping study in response to the research questions are also analyzed in both Sections 5 and 6. Finally, the conclusions and a summary of the state of the art and research opportunities are provided in Section 7.

2. Background

Opdyke [13] stamped the term refactoring and defined what can be considered as the first refactoring catalog. Fowler et al. (Opdyke and Beck as contributors) published the first book on refactoring. According to Fowler's definition, refactoring is "a technique for restructuring an existing body of code, altering its internal structure without changing its external behavior" [14]. The last decade has witnessed a large number of contributions in software refactoring. Refactoring operations were defined, automated and integrated into software development processes. Refactoring opportunities were automatically