Information Systems 50 (2015) 36–50

Contents lists available at ScienceDirect

Information Systems

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

On the usefulness and ease of use of a model-driven Method Engineering approach

Mario Cervera*, Manoli Albert, Victoria Torres, Vicente Pelechano

Centro de Investigación en Métodos de Producción de Software, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

ARTICLE INFO

Article history:
Received 13 December 2014
Received in revised form
15 January 2015
Accepted 17 January 2015
Recommended by D. Shasha
Available online 24 January 2015

Keywords:
Method Engineering
Model-Driven Engineering
Complexity
Think Aloud
Technology Acceptance Model

ABSTRACT

The Method Engineering (ME) discipline emerged as a response to the need for methods that are better adapted to context. Despite the potential benefits of ME and the emergence of Computer-Aided Method Engineering technology, there are hardly any reports on the practical application of ME available in the literature. Some authors argue that this is because practitioners often fail to see the usefulness of ME due to its high complexity. With the aim of facilitating the application of ME, we developed MOSKitt4ME, a lightweight approach that makes intensive use of reusable assets and Model-Driven Engineering. In previous work, we illustrated how MOSKitt4ME supports three phases of the ME lifecycle: design, implementation, and execution. In this paper, we evaluate the complexity of MOSKitt4ME. Specifically, we present a study that is based on the Technology Acceptance Model (TAM) and the Think Aloud method. The TAM allowed us to measure usefulness and ease of use in a subjective manner; the Think Aloud method allowed us to analyze these measures objectively. Overall, the results were favorable. MOSKitt4ME was highly rated in perceived usefulness and ease of use; we also obtained positive results with respect to the users' actual performance and the difficulty experienced.

1. Introduction

Software projects are diverse in nature. They differ, for example, in size, application domain, or team expertise. Due to these differences, it is generally agreed that software companies must define their methods in-house [1–3]; thus, these methods can be adapted to the needs of specific projects. To define methods efficiently and effectively, companies require systematic solutions that are built upon sound methodical foundations. Providing these solutions is the main goal of the Method Engineering (ME) discipline [4]. By adopting ME, companies gain flexibility in building project-specific methods [5,6], and since these methods are defined in-house, developers are motivated to use them due to the feeling of method ownership [7].

Regardless of the potential benefits of ME and the emergence of Computer-Aided Method Engineering (CAME) technology [8], ME has never been widely used in industry [9,10]. Kuhrmann et al. concluded in a recent mapping study [11] that there are hardly any reports on the practical application of ME available in the literature. Henderson-Sellers et al. argue in [2,12] that practitioners often fail to see the usefulness of ME mainly due to its complexity and cost in terms of time, money, and people.

* Corresponding author. Tel.: +34 96 387 70 00x73564.
E-mail addresses: mcervera@pros.upv.es (M. Cervera), malbert@pros.upv.es (M. Albert), vtorres@pros.upv.es (V. Torres), pele@pros.upv.es (V. Pelechano).

http://dx.doi.org/10.1016/j.is.2015.01.006
0306-4379/© 2015 Elsevier Ltd. All rights reserved.

The complexity of ME was also noted by Ter Hofstede et al. [13], who identified several complexity issues related to the selection, storage, retrieval, and assembly of method fragments.

With the aim of facilitating the use of ME, we developed MOSKitt4ME, a ME approach that is fully implemented by a CAME environment [14]. MOSKitt4ME differs from traditional ME in that it is lightweight: MOSKitt4ME is built upon reusability principles and it is also model-driven, which enables a high level of automation. In our previous work [15,16], we illustrated how MOSKitt4ME makes intensive use of reusable assets and Model-Driven Engineering (MDE) to support three phases of the ME lifecycle: the initial design of the method, its implementation, and the final method execution. In this paper, we present an evaluation study that focuses on the complexity of MOSKitt4ME.

The study that is presented in this paper evaluates MOSKitt4ME by means of the Technology Acceptance Model (TAM) [17] and the Think Aloud method [18]. The TAM allowed us to assess the subjective perception of users with respect to two quality attributes: usefulness and ease of use. We evaluated perceived ease of use because this attribute represents a subjective measure of complexity [19,20]. We evaluated perceived usefulness because this attribute is causally affected by perceived ease of use [21], and, for this reason, the usefulness of ME is often negatively perceived by practitioners (which represents a major obstacle for the success of ME and CAME technology). To reinforce the subjective results that were obtained by means of the TAM, we also evaluated usefulness and ease of use in an objective manner. To this end, we analyzed the actual improvement in performance that MOSKitt4ME users achieved during the study and also the difficulties that they experienced.[1] Performance was assessed by measuring efficiency and effectiveness. Difficulty was assessed by analyzing the users' reasoning processes, which reveal the errors made by the users, the doubts that they experienced, and the problem-solving strategies that they followed, among other data. To analyze this data at the highest possible level of detail, we applied the Think Aloud method.

In summary, the contribution of this paper is the thorough evaluation of a model-driven ME approach (MOSKitt4ME) from both a subjective and an objective perspective. The main goal of this evaluation is to illustrate that MOSKitt4ME can be positively rated in terms of perceived usefulness and ease of use and that MOSKitt4ME can also improve the users' performance while posing little difficulty of use. Our positive results contrast with traditional ME, whose usefulness is often negatively perceived by practitioners and whose complexity remains an unsolved issue. As a collateral benefit of the study, we also illustrate how MOSKitt4ME reduces the complexity of ME by means of MDE techniques, which alleviate the users' workload in three phases of the ME lifecycle: design, implementation, and execution.

The remainder of the paper is structured as follows. Section 2 discusses related work and Section 3 summarizes our model-driven ME approach. Then, Section 4 provides an overview of the evaluation study. Each of the four phases that comprise the study are detailed in Sections 5–8. Finally, Section 9 presents some conclusions and outlines future work.

2. Related work

In 1996, Tolvanen et al. [22] noted that ME researchers had focused mostly on the theoretical foundations of the discipline and highlighted the need for investigating usability issues such as usefulness or complexity. A similar conclusion was reached in 1997 by Ter Hofstede et al. [13], who stated that more empirical research was needed to substantiate the claims associated with the potential benefits of ME. Despite these demands for more empirical research, two decades later it is still hard to find empirical studies that investigate methods and tools for ME [11].

One of the few empirical studies that have been conducted in the context of ME is the work by Sousa et al. [23]. This work evaluates the graphical notation of a language for method design: the ISO/IEC 24744 standard [24]. The main contributions of this work are suggestions for improving the notation. Other studies are those by Kelly et al. [25] and Kerzazi et al. [26]. The former evaluates an approach for testing metaCASE environments; this approach is based on an error classification that allows the performance of metamodelers to be measured. The latter evaluates the usability of two method design tools: EPF Composer and DSL4SPM.

In a more theoretical context, we can find two ME approaches that take complexity into consideration. In [9], Bajec et al. present the Process Configuration Approach (PCA), which was conceived to be simple enough to be adopted by software companies. The general idea of the PCA is that project-specific methods are designed by selecting components from a base method. On the other hand, in [3] Karlsson et al. propose the Method for Method Configuration (MMC). The MMC is based on the notion of method component [27], which combines ME with activity theory to make ME less cumbersome.

In addition to the above research efforts, which deal with usability issues, we can also find empirical studies that concern other aspects of ME. For instance, Qumer et al. [28] tested the applicability of a framework for assessing method agility, while in [29] Karlsson describes the lessons learned in the evaluation of a wiki-based approach for method tailoring. On the other hand, Seidita et al. [30] performed a study where they tested their approach for the design of agent-oriented methods.

The analysis of all the aforementioned studies allowed us to identify two important limitations. First, most of the empirical research that has been performed in the ME field only investigates the method design phase of the ME lifecycle; thus, the method implementation and execution phases are almost completely neglected. Second, even though some authors take complexity into consideration [3,9,13], none of them provide a detailed empirical

[1] According to Davis [17], perceived usefulness and perceived ease of use are the people's subjective appraisal of performance and effort/difficulty, respectively.