ABSTRACT

Software companies keep evolving their methods for software production, due to the continuous changes in the organizational, technological, and societal context. Implementing changes to existing methods is a complex activity, which depends not only on understanding ‘what’ to alter, but also on defining ‘how’ to apply the changes. Approaches in the literature are mainly focused on the ‘what’, provide only partial answers to the ‘how’ question, and offer no concrete toolset to support change. In this article, the authors propose and discuss a software toolset that realizes the online method engine (OME) concept and provides a concrete answer to the ‘how’ question by supporting the iterative and incremental assessment, improvement, and enactment of methods for software production. The authors describe the technical architecture of our toolset that concretely realizes our previously proposed OME concept, provide details on the method enactment mechanism, and report on a qualitative evaluation based on interviews with experienced industrial practitioners in process improvement.

Keywords: Experience Management, Incremental Method Engineering, Knowledge Management, Software Process Improvement, Software Product Management

INTRODUCTION

The methods that organizations employ for software production are in constant change, as companies strive to adapt their production processes to new technologies, business models, and development paradigms. Implementing changes in existing methods is a time- and cost-consuming activity, whose success is threatened by multiple factors, including resistance to change, fear of ineffectiveness, and resource constraints (Baddoo, 2003). Unsurprisingly, method change and enactment are important subjects of research in software process im-

DOI: 10.4018/ijismd.2014070101
provement (SPI) (Feiler & Humphrey, 1993; Gonzalez-Perez & Henderson-Sellers, 2008; vom Brocke & Sinnl, 2011).

Much research has been conducted on determining ‘what’ changes to introduce in order to improve process, and thus software, quality. This has resulted in a wide range of frameworks for assessing and improving processes, including the CMMI (CMMI Product Team, 2002) and SPICE (El Emam, 1997) frameworks. While useful, these approaches stay at a high abstraction level and do not answer the important question of ‘how’ to evolve existing methods to accommodate the needed changes.

This research provides an answer to the ‘how’ question from a process-oriented perspective, i.e., by investigating how to support the planning and enactment of changes in software production methods. We do not consider here the equally important cultural or human aspects, such as informing personnel about the change, delivering training sessions, etc. We are motivated by the analysis of software process improvement (SPI) success in small and medium software enterprises by Pino, García, & Piattini (2008), which reveals that two crucial factors are (1) the conduction of an iterative and incremental improvement process that introduces changes based on their priority, and (2) the usage of technical tools and platforms that support the improvement process.

In this paper, guided by these two factors, we describe the realization of a knowledge-based tool that supports the generation of process improvement plans for a specific context. Such plans link the process areas that need to be improved to the available process alternatives and the order in which they are implemented.

Our baseline consists of incremental method engineering (van de Weerd, Brinkkemper, & Versendaal, 2007)—a paradigm that promotes the step-wise improvement of process development processes—and the online method engine (OME) (Vlaanderen, van de Weerd, & Brinkkemper, 2013), the architecture of a knowledge management tool that facilitates incremental process improvement by supporting knowledge dissemination as well as method assessment, improvement, and enactment.

We make the following contributions:

1. We detail the technical toolset that realizes the OME conceptual architecture. To do so, we use the well-known 4+1 architectural view model (Kruchten, 1995).
2. We illustrate how our developed toolset is a sound realization of the OME concept.
3. We detail how we support method enactment through the usage of a widely used, off-the-shelf requirements management tool that includes a workflow engine.
4. We report on a qualitative evaluation of our technical toolset that we obtained through interviews with product managers that have expertise in process improvement.

The remainder of the paper is structured as follows. We first discuss related work, showing a technical toolset for method improvement is needed by researchers and practitioners. We present our baseline and illustrate the challenges to be addressed. We describe the realization of the OME concept by showing structural aspects, the process view, and usage scenarios. We illustrate how we support method enactment through a non-trivial mapping to a workflow engine. We report on empirical evaluation of our approach. Finally, we present conclusions and outline future directions.

RELATED WORK

In software product management (SPM), the main domain of this paper, the skills and practices of a product manager are part of the intellectual capital of a software vendor. However, a problem with this intellectual capital is, as Rus & Lindvall (2002) put it, “it has legs and walks home every day”. This belief is shared by García, Amescua, Sánchez, & Bermón (2011), who tackle the problem by developing a software process knowledge repository developed as a wiki. Similarly, Mishra, Aydin, & Mishra (2013) have developed a basic knowledge manage-
Related Content

Evaluating the Consistency between Navigation and Data Models for Web Applications
[www.igi-global.com/article/evaluating-consistency-between-navigation-data/45926?camid=4v1a](www.igi-global.com/article/evaluating-consistency-between-navigation-data/45926?camid=4v1a)

A Rigorous Approach to the Definition of an International Vocational Master’s Degree in Information Security Management
Frédéric Girard, Bertrand Meunier, Duan Hua and Eric Dubois (2010). *International Journal of Secure Software Engineering* (pp. 1-17).
[www.igi-global.com/article/rigorous-approach-definition-international-vocational/48214?camid=4v1a](www.igi-global.com/article/rigorous-approach-definition-international-vocational/48214?camid=4v1a)
Integrated Software Testing Learning Environment for Training Senior-Level Computer Science Students
www.igi-global.com/chapter/integrated-software-testing-learning-environment/29601?camid=4v1a

What is the Benefit of a Model-Based Design of Embedded Software Systems in the Car Industry?
www.igi-global.com/chapter/benefit-model-based-design-embedded/60727?camid=4v1a