Information and Software Technology 68 (2015) 45–61

Contents lists available at ScienceDirect

Information and Software Technology

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

Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases

Nauman Bin Ali [a,*], Kai Petersen [a], Breno Bernard Nicolau de França [b]

[a] Blekinge Institute of Technology, Karlskrona, Sweden
[b] ESE Group, PESC/COPPE, Federal University of Rio de Janeiro, Rio de Janeiro 68511, Brazil

ARTICLE INFO

Article history:
Received 3 February 2015
Revised 17 August 2015
Accepted 17 August 2015
Available online 29 August 2015

Keywords:
Case study research
Value stream mapping
Lean software development
Software process simulation modeling
System dynamics

ABSTRACT

Context: Value stream mapping (VSM) as a tool for lean development has led to significant improvements in different industries. In a few studies, it has been successfully applied in a software engineering context. However, some shortcomings have been observed in particular failing to capture the dynamic nature of the software process to evaluate improvements i.e. such improvements and target values are based on idealistic situations.

Objective*:* To overcome the shortcomings of VSM by combining it with software process simulation modeling, and to provide reflections on the process of conducting VSM with simulation.

Method*:* Using case study research, VSM was used for two products at Ericsson AB, Sweden. Ten workshops were conducted in this regard. Simulation in this study was used as a tool to support discussions instead of as a prediction tool. The results have been evaluated from the perspective of the participating practitioners, an external observer, and reflections of the researchers conducting the simulation that was elicited by the external observer.

Results: Significant constraints hindering the product development from reaching the stated improvement goals for shorter lead time were identified. The use of simulation was particularly helpful in having more insightful discussions and to challenge assumptions about the likely impact of improvements. However, simulation results alone were found insufficient to emphasize the importance of reducing waiting times and variations in the process.

Conclusion: The framework to assist VSM with simulation presented in this study was successfully applied in two cases. The involvement of various stakeholders, consensus building steps, emphasis on flow (through waiting time and variance analysis) and the use of simulation proposed in the framework led to *realistic* improvements with a high *likelihood* of implementation.

1. Introduction

Enthralled by the success of lean development in manufacturing and product development [1] and the service industry [2], lean concepts have been adapted to the area of software development [3]. One of the practices in lean software development is value stream mapping (VSM) [3]. VSM is the practice of creating a value stream map that identifies the value added by each step in the software development process [4].

Starting with the *current state map* (CSM), this practice identifies *value-adding, required non-value adding*, and *non-value adding* activities in the current process. Making this distinction enables the improvement in the current process by the elimination of non-value adding activities [4] and creates a desired *future state map* (FSM) and an action plan for improvements.

VSM is a practice that *"straddles the gray area between"* concrete practices and analytical principles of *Lean* [5]. It implements several of Lean principles directly (like "optimize the whole", "eliminate waste") and contributes to fulfilling many others (like "continuous improvement", "flow" and "pull-based" development) [6,7]. For example, VSM puts to use the principle *"optimize the whole"* by taking several measures to take a system-wide perspective. It does so by considering the end-to-end process and involving multiple stakeholders responsible for various activities in the process [8], both in

* Corresponding author. Tel.: +46 455385541.
E-mail addresses: nauman.ali@bth.se (N.B. Ali), kai.petersen@bth.se (K. Petersen), bfranca@cos.ufrj.br (B.B.N. de França).

http://dx.doi.org/10.1016/j.infsof.2015.08.005
0950-5849/

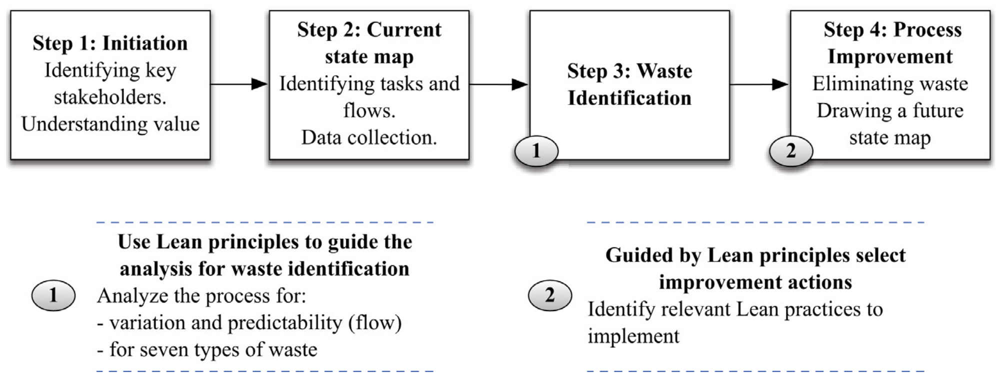

Fig. 1. Overview of the VSM steps and their role in the operationalization of Lean principles.

identification of waste and improvements. This ensures that the focus is on the most significant impediments hindering the organization from delivering value to the customer.

Moreover, as our understanding of customer value improves we can perform VSM to realign the development process to continue to deliver value efficiently to the customer. This provides the means necessary for *"continuous improvement"*. VSM helps to identify waste by visualizing and analyzing the current value stream. This helps to operationalize another principle of *Lean* that is to *"eliminate waste"*. The types of waste (e.g. partially done work, unnecessary features) listed by Poppendieck and Poppendieck [6] should only be considered as a guideline however, it is the VSM activity that will help to see where these manifest in the concrete case of a company's process and which types of waste should be handled with priority.

Fig. 1 depicts how VSM can systematically operationalize the Lean principles by using them first to guide the analysis of the current value stream, then in identification of waste and lastly in identifying which improvements to implement.

The use of VSM has led to several tangible benefits e.g. McManus [9] reports studies where organizations significantly reduced the standard deviation of the cycle times, the rework required to fix defects and the cycle time by 75%. It is also reported that typically up to 25% savings in engineering efforts are achieved with VSM based improvements [9].

Similar to manufacturing processes, in the development process for aerospace, information inventories had *"engineering work-packages"* inactive for up to 77% of the time [9]. The success of VSM outside manufacturing and production and in the design and development process of engineering makes it interesting to explore if it can be leveraged in software development.

Khurum et al. [8] adapted and applied VSM in the context of large-scale software intensive product development. They found the following two shortcomings based on practitioners feedback [8]. First, the notation used only provides a snapshot of the system and fails to capture the dynamic aspects of the underlying processes. This leads to a simplistic analysis to identify bottlenecks. Second, the improvement actions and the target value maps are based on idealistic situations. This also reduces the realism in the target value stream map (as it is based on ideal conditions that the identified problems are completely resolved).

These limitations reduce the confidence in the improvement actions identified in VSM, making it less likely that such improvement actions will be implemented. Given these limitations [8], and the potential benefits of software process simulation modeling (SPSM) [10], we argue that SPSM can be used to overcome these shortcomings. Other disciplines where VSM has been used have also seen a utility in combining VSM with simulation (cf. [11–16]). Some reasons for such combination in these disciplines are listed below:

- Simulation helps to reason about changing the process, and supports consensus building by visualizing dynamic views of the process [11,12,17–25].
- VSM alone is time consuming and simulation can assist to speed-up the analysis of the CSM, and the derivation and verification of a FSM [25–28]. Simulation helps to verify the impact of changes and answering questions that cannot be answered by VSM [13,28–30].
- VSM provides a snapshot, it is unable to detail dynamic behavior. Simulation can help predict the flow and levels and provide more accurate quantitative measures given its ability to handle both deterministic and stochastic inputs [16,17,26,31].
- VSM alone cannot capture the complexity in terms of the iterations, overlaps, feedback, rework, uncertainty and stochasticity of the process [14,26,32–35].

The contributions of this study can be briefly summarized as follows:

Contribution 1: Based on literature, this study proposes a framework to perform simulation-assisted VSM.

Contribution 2: Evaluation of the use of VSM and the simulation support in industry.

To achieve the evaluation contribution, we conducted the study at a development site of Ericsson AB, Sweden. This entailed 10 (3 h long) workshops for two products with five to six relevant practitioners in each workshop, and extensive quantitative and qualitative data gathering and analysis.

The evaluation of simulation-assisted VSM was done from three different perspectives, namely: (*a*) the practitioners participating in the activity, who are to benefit from and utilize the approach (*b*) the facilitators (first and the second author of the article) who conducted the simulation-assisted VSM (*c*) an external observer (third author of the paper) who is a simulation expert being present during the workshops. The external observer assesses the process from an academic point of view. His expertise (having defined guidelines for simulation and prior hands-on simulation experience) allowed a critical observation of our use of simulation in a practical situation. He had a guideline focus compared to the practitioners, facilitators and our view as researchers doing applied research using simulation in the context of VSM.

Furthermore, we provide a detailed account of the findings to illustrate which wastes and improvements could be identified through this approach.

The remainder of the paper is structured as follows. Section 2 presents related work. Section 3 maps the strategies for combining simulation with VSM and details the process for performing it (Contribution 1). Section 4 describes the research method.