

# Chapter 1

## Process and Productivity Improvement in Agile Software Development with Process Libraries: Case Study

**Hugo A. Mitre**

*Research Center in Mathematics (CIMAT), Mexico*

**Leonardo Bermon-Angarita**

*National University of Colombia, Colombia*

### **ABSTRACT**

*Currently, agile methods are replacing traditional process-based methods in the software industry. However, process-based software development still matters because of its degree of reusability in new projects. Some problems arise when Knowledge Management (KM) is not correctly aligned with processes, such as lack of productivity and process improvement. In this chapter, the authors present a case study of applying two proposals of process libraries in an agile software development division. The researchers and the software development division worked with a Kanban dashboard, eXtremme Programming (XP), and SCRUM practices, adapted to Process Assets Library (PAL) and to Process Practice Library (PPL) in three software projects in order to find experiences of tacit and explicit knowledge that have an impact on process improvement and productivity. Under results and discussion, the authors present the good and bad practices.*

DOI: 10.4018/978-1-4666-5182-1.ch001

## 1. INTRODUCTION

During these last years, agile methods are becoming more important and rapidly replacing traditional methods for software development companies (Laanti, Salo, & Abrahamsson, 2011). But this does not mean the demise of software development based on processes. The time and effort to create a project process description is significantly reduced when a reusable process is instantiated instead of building the process from scratch (Hollenbach & Frakes, 1996).

One concern for creating a process description is that software developers do not usually follow the process to search and create knowledge required to develop and improve their activities (Jean-Claude & Oquendo, 2004). There are two types of knowledge leading the software development process: knowledge acquired through experience (known as tacit knowledge), and the technical knowledge stored in documents (known as explicit knowledge). Nonaka (1994) describes explicit knowledge as stored in textbooks, software products and documents, and implicit (tacit) knowledge as stored in the minds of people in the form of memory, skills, experience, education, imagination and creativity. One way to organize knowledge is to use and manage repositories of process assets known as a Process Asset Library (PAL). A PAL captures, understands and uses processes to improve productivity (Jalote, 2002).

Since the information on agile software development is easy to obtain, one can say that explicit knowledge is less worrisome than tacit. Also, there are benefits from applying practices derived from experiences - tacit knowledge - to agile methods, such as the improvement of customer collaboration, fixing problems while work is in progress and learning with pair programming (Dybå, Dingsøy, & Moe, 2004). Other important practices of agile methods are stand-up meetings (Levy & Hazzan, 2009) and iteration retrospective. However, there is no evidence of productivity and

process improvement of knowledge management having an impact on agile methods in a real case study. Our research goal is to:

*Find experiences of how the use of tacit and explicit knowledge has an impact on productivity in agile software development and process improvement.*

In this article, we provide evidence of process improvement and productivity from agile software development teams. For the case study, we used two process libraries: PAL was applied to two small projects and the PPL (Process Practice Library) to a small project in a software company. PPL was adapted and improved in the agile software process and the technical needs of the software company.

This article is organized as follows: section two presents the PAL and PPL proposals; section three describes the case study; section four analyzes and discusses the results; and the last section presents the conclusions.

## 2. PROCESS LIBRARIES

Before this proposal, there were two process-oriented libraries: the first was a Web-based library oriented towards small software companies to provide organized and searchable repository of process assets and to cover the needs of process guidance for the Capability Maturity Model (CMMI) (Calvo-Manzano, Cuevas, & San Feliu, 2008). The second was a wiki-based library (Wiki 2.0 offering dynamic content) oriented towards small and medium software companies with the aim of searching, creating and modifying the tacit and explicit knowledge of process assets, called PAL (García, Amescua, Sánchez, & Bermón, 2011). A process asset can help to understand, capture, and use the process. Moreover, the assets generated in previous projects can be stored in the wiki at the end of the projects and reused in future projects. Reuse of these assets can save

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/process-and-productivity-improvement-in-agile-software-development-with-process-libraries/100267](http://www.igi-global.com/chapter/process-and-productivity-improvement-in-agile-software-development-with-process-libraries/100267)

## Related Content

---

### Multiple Multimodal Mobile Devices: Lessons Learned from Engineering Lifelog Solutions

Daragh Byrne, Liadh Kelly and Gareth J.F. Jones (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications* (pp. 706-724).

[www.irma-international.org/chapter/multiple-multimodal-mobile-devices/66494](http://www.irma-international.org/chapter/multiple-multimodal-mobile-devices/66494)

### Model-Driven Performance Evaluation of Web Application Portals

Nilabja Roy and Douglas C. Schmidt (2011). *Model-Driven Domain Analysis and Software Development: Architectures and Functions* (pp. 407-437).

[www.irma-international.org/chapter/model-driven-performance-evaluation-web/49169](http://www.irma-international.org/chapter/model-driven-performance-evaluation-web/49169)

### An Exploratory Study of the Security Design Pattern Landscape and their Classification

Poonam Ponde and Shailaja Shirwaikar (2016). *International Journal of Secure Software Engineering* (pp. 26-43).

[www.irma-international.org/article/an-exploratory-study-of-the-security-design-pattern-landscape-and-their-classification/160711](http://www.irma-international.org/article/an-exploratory-study-of-the-security-design-pattern-landscape-and-their-classification/160711)

### Improving Software Design by Mitigating Code Smells

Randeep Singh, Amit Kumar Bindal and Ashok Kumar (2022). *International Journal of Software Innovation* (pp. 1-21).

[www.irma-international.org/article/improving-software-design-by-mitigating-code-smells/297503](http://www.irma-international.org/article/improving-software-design-by-mitigating-code-smells/297503)

### Private Blockchain-Cloud System (PBCS) for Healthcare Services

Sangeeta Gupta (2022). *International Journal of Software Innovation* (pp. 1-8).

[www.irma-international.org/article/private-blockchain-cloud-system-pbcs-for-healthcare-services/289602](http://www.irma-international.org/article/private-blockchain-cloud-system-pbcs-for-healthcare-services/289602)