

Danilo Caivano
Markku Oivo
Maria Teresa Baldassarre
Giuseppe Visaggio (Eds.)

Product-Focused Software Process Improvement

12th International Conference, PROFES 2011
Torre Canne, Italy, June 2011
Proceedings

LNCS 6759

Caivano et al. (Eds.)



LNCS
6759

Product-Focused Software Process Improvement

Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at www.springer.com/lncs

Proposals for publication should be sent to

LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany

E-mail: lncs@springer.com

ISSN 0302-9743

ISBN 978-3-642-21842-2



Lecture Notes in
Computer Science

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

- David Hutchison
Lancaster University, UK
- Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
- Josef Kittler
University of Surrey, Guildford, UK
- Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
- Alfred Kobsa
University of California, Irvine, CA, USA
- Friedemann Mattern
ETH Zurich, Switzerland
- John C. Mitchell
Stanford University, CA, USA
- Moni Naor
Weizmann Institute of Science, Rehovot, Israel
- Oscar Nierstrasz
University of Bern, Switzerland
- C. Pandu Rangan
Indian Institute of Technology, Madras, India
- Bernhard Steffen
TU Dortmund University, Germany
- Madhu Sudan
Microsoft Research, Cambridge, MA, USA
- Demetri Terzopoulos
University of California, Los Angeles, CA, USA
- Doug Tygar
University of California, Berkeley, CA, USA
- Gerhard Weikum
Max Planck Institute for Informatics, Saarbruecken, Germany

Danilo Caivano
Markku Oivo
Maria Teresa Baldassarre
Giuseppe Visaggio (Eds.)

Product-Focused Software Process Improvement

12th International Conference, PROFES 2011
Torre Canne, Italy, June 20-22, 2011
Proceedings

Preface

Volume Editors

Danilo Caivano
University of Bari, Department of Informatics
Via E. Orabona 4, 70126 Bari, Italy
E-mail: caivano@di.uniba.it

Markku Oivo
University of Oulu, Department of Information Processing Science
P.O. Box 3000, 90014 Oulu, Finland
E-mail: markku.oivo@oulu.fi

Maria Teresa Baldassarre
University of Bari, Department of Informatics
Via E. Orabona 4, 70126 Bari, Italy
E-mail: baldassarre@di.uniba.it

Giuseppe Visaggio
University of Bari, Department of Informatics
Via E. Orabona 4, 70126 Bari, Italy
E-mail: visaggio@di.uniba.it

ISSN 0302-9743
e-ISSN 1611-3349
ISBN 978-3-642-21842-2
e-ISSN 978-3-642-21843-9
DOI 10.1007/978-3-642-21843-9
Springer Heidelberg Dordrecht London New York
Library of Congress Control Number: 2011929368

CR Subject Classification (1998): D.2, K.6, J.1, H.3-4, C.2.4, J.3
LNCS Sublibrary: SL 2 – Programming and Software Engineering

© Springer-Verlag Berlin Heidelberg 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services Chennai, India

On behalf of the PROFES Organizing Committee we are proud to present the proceedings of the 12th International Conference on Product-Focused Software Process Improvement (PROFES 2011), held in Torre Canne, Italy. Since 1999 PROFES has grown in the software engineering community and has become a premium conference that brings together both academia and industry.

The roots of PROFES lie in the professional software process improvement motivated by product, process and service quality needs. The conference retains its high quality and focus on the most relevant research issues by addressing both perspectives, research and practice, from an academic and industrial point of view.

Today's software products and services are perceived as strategic assets for empowering business sectors at every level of the value chain, from strategic to operative. In this scenario, and considering the current global economic downturn, the challenge for developing software products and services consists in managing process diversity in order to reuse strategic software assets in various fields and environments quickly and cost effectively. This was the special theme for PROFES 2011.

In the last few years, many approaches and techniques have been proposed for managing diversity: experience bases for collecting and sharing knowledge and experiences; software development processes able to rearrange common assets in diverse products; process patterns as an instrument for filling the gap between process definition and the amount of customizations needed; estimation and calibration techniques that deal with the different processes in use; parametric and goal-oriented quality models; project management techniques able to fulfil the project goals in spite of project characteristics; cloud computing and service orientation for managing the diversity of hardware and software platforms. All these innovations provide exciting opportunities to make significant progress in understanding and facing real-world challenges.

This year's technical program featured invited talks, research papers, and experience reports on the most relevant topics in the focus area. We received 54 papers submitted from 22 nations, with each paper receiving at least three reviews. After a thorough evaluation, the Program Committee selected 24 technical full papers. The topics addressed in these papers indicate that the PROFES theme is a vibrant research area, but is also of high interest for industry as demonstrated by several papers that report on case studies or experience gained in industry.

We were proud to have two top keynote speakers: (1) Dennis Smith – Senior Member of the Technical Staff at Carnegie Mellon University's Software Engineering Institute and (2) David I. Foster – Senior Technical Fellow, View

short-paper session, two workshops, Managing the Client Value Creation Process in Agile Projects (VALOIR) and Project and Knowledge Management Trends (PKMT), one tutorial, Establishing and Improving Project Management Using Assessment Models for Process Capability and Organizational Maturity, and a Doctoral Symposium.

We wish to thank the University of Bari, the University of Oulu, the competence center Driving Advances of ICT in South Italy-Net (DAISY-NET), the Project Management Institute Southern Italy Chapter (PMI-SIC) and Software Engineering Research and Practices s.r.l. (SER&Practices) – spin-off of the University of Bari – for supporting the conference. We are also grateful to the authors for their high-quality papers and the Program Committee for their hard work in reviewing the papers.

June 2011

Daniilo Caivano
Markku Oivo
Maria Teresa Baldassarre
Giuseppe Visaggio

Organization

PROFES 2011 was organized by the University of Bari and University of Oulu.

General Chair

Giuseppe Visaggio
University of Bari and Daisy-Net, Italy

Program Co-chairs

Daniilo Caivano
Markku Oivo
University of Bari and SER&Practices, Italy
University of Oulu, Finland

Organizing Chair

Maria Teresa Baldassarre
University of Bari and SER&Practices, Italy

Publicity Chair

Matias Vierimaa
VTT, Oulu, Finland

Publicity Co-chair Asia

Katsuro Inoue
Osaka University, Japan

Publicity Co-chair Australia

Emam Hossain
NICTA, Australia

Publicity Co-chair Europe

Luigi Buglione
ETS / Engineering Group, Italy

Publicity Co-chair South America

Guilherme H. Travassos
COPPE/UF RJ Federal University of Rio de Janeiro,
Brazil

Short Papers and Poster Chairs

Guilherme H. Travassos
COPPE/UF RJ Federal University of Rio de Janeiro.

Doctoral Symposium Chairs

Marcela Genero
Emilia Mendes

University of Castilla la Mancha, Spain
University of Auckland, New Zealand

Workshops and Tutorials Chairs

Alberto Sillitti
Felix Garcia

Free University of Bolzano, Italy
University of Castilla la Mancha, Spain

Web Chair

Giovanni Bruno

University of Bari, Italy

Program Committee

Zeid Abdelnabi

Garyounis University - IT College, Libya
Universidad Politécica de Valencia, Spain

Silvia Abrahão

ITU of Copenhagen, Denmark

Muhammad Ali Babar

University of Bari, Italy

Pasquale Ardimento

University of Bari, Italy

Maria Teresa Baldassarre

Technical University of Vienna, Austria

Stefan Biffi

SWPM - Software.Process.Management, Germany

Andreas Birk

ETS / Engineering Group, Italy

Luigi Buglione

University of Bari, Italy

Danilo Caivano

University of Sannio, Italy

Gerardo Canfora

Alabama University, USA

Jeffrey Carver

Fraunhofer Institute for Experimental Software Engineering, Germany

Marcus Ciolkowski

University of Bristol, UK

Dave Cliff

Norwegian University of Science and Technology, Norway

Reidar Couradi

Second University of Naples, Italy

Beniamino Di Martino

SINTEF, Norway

Torgeir Dingsøy

SINTEF, Norway

Tore Dybå

University of Rome "Tor Vergata", Italy and Simula, Norway

Davide Falessi

Fraunhofer Center Maryland, USA

Raimund Feldmann

University of Szeged (SZTE), Hungary

Rudolf Ferenc

University of Calabria, Italy

Alfredo Garro

Johannes Kepler University Linz, Austria

Paul Gruenbacher

Fraunhofer Institute for Experimental Software Engineering, Germany

Jens Heidrich

Engineering, Germany

Frank Hordab

Hajimu Iida
Michel Jaring
Natalia Juristo
Janne Järvinen
Yiannis Kanellopoulos

NAIST, Japan
Universitiy of Helsinki, Finland
Universidad Politécica de Madrid, Spain
F-Secure, Finland
Software Improvement Group, Amsterdam, The Netherlands

Petri Kettunen
Pasi Kuvaaja
Lech Madeyski

University of Helsinki, Finland
University of Oulu Finland

Kenichi Matsumoto

Wroclaw University of Technology, Poland
Nara Institute of Science and Technology, Japan

Makoto Matsushita

Osaka University, Japan

Maurizio Morisio

Politecnico di Torino, Italy

Mark Mueller

Robert Bosch GmbH, Germany

Juergen Muench

Fraunhofer IESE, Germany

Haruka Nakao

Japan Manned Space Systems Corporation, Japan

Risto Nevalainen

FiSMA ry, Finland

Mahmood Niazi

Keele University, UK

Linda Northrop

Software Engineering Institute, Carnegie Mellon University, USA

Markku Oivo

University of Oulu, Finland

Paolo Panaroni

INTECS, Italy

Dietmar Pfahl

Lund University, Sweden

Minna Pikkarainen

VTT, Finland

Teade Punter

Embedded Systems Institute (ESI), Netherlands

Austen Rainer

University of Hertfordshire, UK

Daniel Rodriguez

University of Alcalá, Spain

Barbara Russo

Free University of Bolzano-Bozen, Italy

Outi Salo

Nokia, Finland

Klaus Schmid

University of Hildesheim, Germany

Kurt Schneider

Leibniz Universität Hannover, Germany

Michael Stupperich

Daimler AG, Germany

Christos Tjortjis

Univ. of Ioannina, Greece and University of W. Macedonia, Greece

Guilherme Travassos

COPPE/UFRJ, Brazil

Markku Tukiainen

University of Joensuu, Finland

Mark Van Den Brand

Eindhoven University of Technology, The Netherlands

Rini Van Solingen

Delft University of Technology, Netherlands

Sira Vegas

Universidad Politecnica de Madrid, Spain

Aaron Visaggio

University of Sannio, Italy

Hironori Washizaki

National Institute of Informatics, Japan

X Organization

Additional Reviewers

A

Acuña, Silvia T.

D

Dieste, Oscar

F

Fernandez, Adrian

Fgri, Tor Erlend

Ficco, Massimo

Fushida, Kyohei

G

González-Huerta, Javier

H

Haugset, Brge

Hegedús, Péter

K

Kakuja-Tóth, Gabriella

Klabbers, Martijn

M

Masscollar Da Silva, Jobson Luiz

N

Nicolau De França, Breno Bernard

O

Ohkura, Kimiharu

S

Scheinoltz, Lauri Ann

Y

Yoshida, Norihiro

Table of Contents

Keynote Addresses

The Impact of Emerging Software Paradigms on Software Quality and User Expectations.....	1
<i>Dennis B. Smith</i>	
Acquiring Information from Diverse Industrial Processes Using Visual Analytics	2
<i>David J. Kasik</i>	

Agile and Lean Practices

Monitoring Bottlenecks in Agile and Lean Software Development Projects – A Method and its Industrial Use	3
<i>Miroslaw Staron and Wilhelm Meding</i>	
Applying Agile and Lean Practices in a Software Development Project into a CMMI Organization	17
<i>Miguel Morales Trujillo, Hanna Ohtaba, Francisco J. Pino, and María J. Orozco</i>	
Adopting Agile Practices in Teams with No Direct Programming Responsibility – A Case Study	30
<i>Kirsi Korhonen</i>	

Cross-Model Quality Improvement

Proposing an ISO/IEC 15504-2 Compliant Method for Process Capability/Maturity Models Customization	44
<i>Jean Carlo Rossa Hauck, Christiane Gresse von Wangenheim, Fergal Mc Caffery, and Luigi Buglione</i>	
Homogenization, Comparison and Integration: A Harmonizing Strategy for the Unification of Multi-models in the Banking Sector	59
<i>César Pardo, Francisco J. Pino, Félix García, Mario Piattini, María Teresa Baldassarre, and Sandra Lemus</i>	
Supporting Audits and Assessments in Multi-model Environments	73
<i>Andre L. Ferreira, Ricardo J. Machado, and Mark C. Paulk</i>	

Applying Agile and Lean Practices in a Software Development Project into a CMMI Organization

Miguel Morales Trujillo¹, Hanna Oktaba¹, Francisco J. Pino^{2,3}, and María J. Orozco⁴

¹ Graduate Science and Engineering Computing, National Autonomous University of Mexico
Mexico City, Mexico

(migmor, hanna.oktaba)@ciencias.unam.mx

² IDIS Research Group

Electronic and Telecommunications Engineering Faculty, University of Cauca

Calle 5 No. 4-70, Popayan, Colombia

fjpino@unicauca.edu.co

³ Alarcos Research Group

Institute of Information Technologies & Systems, University of Castilla-La Mancha

Paseo de la Universidad 4, Ciudad Real, Spain

francisco.pino@uclm.es

⁴ Ultrasist

Miguel Ángel 28, Mexico City, Mexico

mjorozcom@ultrasist.com.mx

Abstract. This paper presents an approach based on a practical experience in applying agile and lean practices in a software development process performed into an organization evaluated CMMI level 5. As a result of a theoretical review on agile and lean practices, and the organization's needs, an integrated proposal between these practices and CMMI was found and was also put into practice. The work carried out by the organization using this proposal led to a successful integration experience in order to innovate, improve product quality, get clients' satisfaction, and the most important, show the feasibility of coexisting of CMMI and agile practices resulting in a significant improvement for the organization.

Keywords: Agile practices, Lean practices, SCRUM, CMMI, Software Development Process.

1 Introduction

In a software developer organization, the need to improve their capabilities is mandatory. One way to reach that goal is to adopt the best practices from a process reference model which guides the organizational improvement.

A world known reference model is the Capability Maturity Model Integrated (CMMI), it provides an approach to improvement in which processes and improvements to them are supported by the organization [1].

It is a strong belief that agile development methods and CMMI best practices are orthogonal with each other. The experience presented in this work is intended to

clarify and highlight the benefit from using both of them resulting into an improved business performance, as has been suggested by [1]. Furthermore, according to [2] the agile methods can support some success factors involved in improving organizational processes.

The main contribution of the practical experience discussed in this paper is to show how an organization assessed CMMI level 5 has integrated their software development process with agile and lean practices in order to inject agility, simplicity and refreshment into the processes, increasing the quality of the software product and the client's satisfaction. In this respect, this work reports a successful experience of melting agile practices into a CMMI level 5 organization. However it is important to highlight that it is not the purpose to present a new model or methodology, neither to create a new agile method. Clearly speaking, our approach is based on a selection of agile and lean practices suitable for a particular project in order to improve the processes, but not to replace it.

The paper is organized as follows: in part 2 we present the background, part 3 shows the context of the organization and the project that we present as experience. The experience report is shown in part 4, and finally we conclude and mention the future work.

2 Background

Several solutions to improve software development processes are suggested, many of these suggestions have come from experienced practitioners, who have labeled their methods as *agile software development* [3].

Agile methods have been demonstrated to enhance customer value, improve organizational morale, and provide step improvement in product quality [4]. For that reason numerous organizations have chosen this particular path to improve; however the big problem is to find the right balance between agility and discipline.

According to [5], for a company in transition towards agile software development there is a need for practices and guidance for implementing and supporting an agile approach across the organization. Besides, taking into account the organizational context helps eliminate the suspicion about agility [6].

Another area of conflict for mature organizations will be the problem of considering how or to what extend agile processes will affect their ratings with respect to CMMI, ISO, or other process standards [7]. However, agile methods are in line with many CMMI level 5 concepts, mainly regarding continuous improving of processes performance, thus, instead of acting like a barrier, the agile methods impulse those CMMI concepts.

On the other hand, talking about their limitations, most agile methods do not support the degree of documentation and infrastructure required for lower-level certification; it might, in fact, make agile methods less effective [7].

From that point of view, the difficulty stands in filling the gap between the traditional development and the agile development, for that reason it is a favorable alternative to try to transform or remove particular aspects of the processes to make the most of each one. Lean software development [8] [9] promotes removing waste as one of

its principles. According to [10], extra or unnecessary functionality, slow internal communication, task switching and bureaucracy fall into the category of "waste". However, complexity science seems to show that waste can have various functions; in complex systems things that look like waste can actually be a source for stability and innovation [11].

In order to preserve stability into the processes, approaches like [7] suppose to help the integration of agile practices into the traditional process, firstly, by conducting an analysis of existing and proposed processes to identify mismatches in process requirements and expectations. Secondly, by taking into account the project's needs, there builds up a process with indispensable components, and specific responsibilities are defined to address with the agile approach, establishing the milestones to better fit an iterative approach.

Later on, the integration continues by implementing agile practices that support existing processes or new organizational priorities, like prioritizing requirements to keep on a schedule when new requirements emerge, or test-first and continuous integration to find problems beforehand.

An improvement seeking organization wants to assess the impact of process changes before introducing them to improve the way of working, however in development projects the change proposal is evaluated in a real situation [21]. A case of study is suitable for industrial evaluation of software engineering methods, according to [21], one way to arrange a case study, it's to make a comparison of results of using a new method against a company base-line, then is possible to compare both.

In this context, to develop the projects reported in this paper, which seek to incorporate agile and lean practices in the development process of a CMMI level 5 organization, the approach mentioned above was taken into account. This work reports the qualitative experience of transitioning from the traditional development to the agile development, by means of two projects carried out for this purpose, within an organization that has used CMMI as reference model to define its software development life cycle. Likewise it presents challenges faced by the project teams, as well as the benefits of using agile practices, and practical recommendations for those starting and facing similar challenges.

3 Context of the Software Development Project

This work was developed at Ultrastit [12], a Mexican organization certified with CMMI level 5 version 1.2 since March 2009. Ultrastit is oriented to develop applications through a work environment based on processes and international standards of quality. At the time of the experience presented, the organization was made up by 83 employees.

Innovation is a main objective of the organization, for that reason it created a weekly workshop called *Research Seminar*, where the members propose new ideas, methodologies, solutions and improvement opportunities in order to optimize the group of processes and performance of the organization.

One of the concerns at the workshop was introducing agile practices into the process with the objective of improving particular aspects. The existence of agile practices at the organization was not new, but they were merely focused on internal processes and activities, for example, as a result of this workshop, a technique based on Pair Programming and called eXtreme Analysis was developed. It is executed during the Analysis phase in order to improve the definition of the requirements, alongside with carrying out the Peer Review in every phase.

In this case, three factors were identified to be improved: customer's satisfaction, work team communication and, the most ambitious, reduction of the release time of a product.

At that moment a new project was required by a frequent client; thus, the High Management approved to search a set of practices that could be applied by the work team assigned to that project. The software products associated with those projects were related to receipt, control and administration of funds. For privacy reasons we will not mention the client's name and it will be simply referred to as "the client" and the projects involved as "PAL" (acronym for *Projecting Agile & Lean*).

After the ending of PAL1, the client asked for another product and we will call the corresponding project as PAL2. In general terms, when we mention PAL projects we will be referring to both PAL1 and PAL2 projects.

In order to speed up PAL projects, the organization assigned two resources (PAL team) from the Process Group, one of them being the first author of this paper. The PAL team was given the task to identify specific processes or activities candidates to be replaced or improved by agile and lean practices. Subsequently, those resources developed the necessary material, on one hand to apply the selected practices into the development process and, on the other hand, to train the work team to apply the required knowledge and achieve the desired objectives.

3.1 Project Description

The PAL projects involved the phases of Analysis, Design, Construction and Tests together with the resources specialized in each of those phases. The organization manages its resources in two levels: horizontal and vertical. The horizontal level consists in a group controlled by a specialist called *Technical Leader*, for example, the Specification Leader manages all the people with the role of Analyst, and the Test Leader is in charge of all the testers.

At the vertical level the main role is played by the Project Manager who is in charge of a team made up by people working on a particular project. Therefore, a person, who has issues associated with the technical part, should inform their Technical Leader; on the other hand, if he has issues associated with the management of the project, he should contact the Project Manager in charge of that project.

The nature of the projects required developing of all the Testing phases at the client's facilities. For that reason there have existed historically communication issues between the Test leader who stays at Ultrasist and the testers' team that has to be at the client's, as well as between the Construction leader and the programmers' team.

The work team assigned to the projects was composed by 7 members: 1 Project Manager, 2 Analysts, 1 Designer, 2 Programmers and 1 Tester, where it is important to observe that those are the official roles that they play in the organization normally.

However, for the purpose of the PAL projects their functions and responsibilities were not limited to those roles. The size of each project was given in use-case points, being 235 use-case points for PAL1 and, 270 for PAL2, respectively. The estimated delivery dates were for PAL1 March 19th totalizing 35 estimated working days and, for PAL2 May 14th totalizing 40 working days, both developed during 2010. See Table.

Table 1. Estimated dates, use-case points and hours of each project

Project name	Contract Dates		Use-case points	Effort (in hours)
	Beginning	Closing		
PAL 1	February 1 st	March 19 th	235	603.50
PAL 2	March 22 nd	May 14 th	270	628.45

The client's satisfaction was a hard issue to deal with because of their high demand and low commitment, with almost none chances for communication after the contract is signed and ever-changing requirements on road.

4 Experience Report

To perform the introduction of agile and lean practices into the organization processes, the PAL team in the first place identified the specific processes or activities necessary to be replaced or improved by agile and lean practices.

Later it developed the necessary material to apply the selected practices into the development process and finally introduced the work team to the new practices. The PAL team work is described below to more detail.

4.1 Project Execution

In order to make clearer the structure and the execution, the project was divided into five steps. Each step having a specific purpose, the first four steps were developed only by the PAL team, and the last one developed in conjunction with the work team.

Step 0 had as its objective the selection of agile and lean concepts. Those concepts were selected mainly from SCRUM [13], Lean and Test Driven Development [14]. Some of the concepts selected were:

- Self directed and self organized team.
- Daily meetings are carried out to listen to the progress feedback and to troubleshoot impediments.
- Short iterations.
- Quantifiable measurements of progress.
- Design for change and respond to change.
- Bring decisions forward, Planning and responding.
- Design test first and test in short cycles.
- Fuzz the distinction between requirements and tests.

SCRUM was chosen as based on the approach described in [15], where an adaptation of SCRUM to a traditional software development process is presented.

Additionally, the SCRUM process flow, see Figure 1, provides guidance for efficient project management in a certain way that allows high flexibility and adaptability, and CMMI provides insight into what processes are needed to maintain discipline [16]. Thus, an accurate mix is obtained from both of them to start the improvement focused on the main objectives.

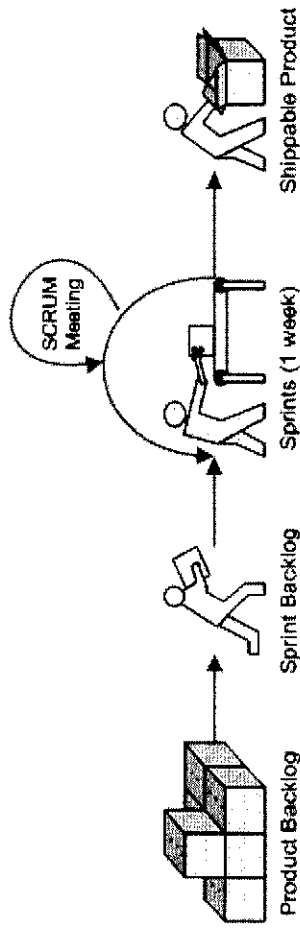


Fig. 1. SCRUM process flow

At Step 1, there was created a cookbook according to the organization processes and the agile and lean practices stemmed from the concepts selected in Step 0. Moreover, a couple of presentations and a guide were produced to introduce those selected items to the work team during the next steps. One presentation was based fully on SCRUM and its principles, benefits, requirements and practices for project management. The other presented the objectives of the organization that pursue agility and lean introduction.

The cookbook was divided into three parts, where the first one describes the actual software development process of the organization and the candidate practices to be replaced or improved; the next section shows the agile and lean practices selected, mainly for the project management and the test phase; and the last one presents the method of introduction and application of these new practices, including activities, roles customization and guidelines for tailoring the process. In Table 2 a fragment of the cookbook is shown.

The cookbook's contents focus on three major aspects: Planning: arrangement of the activities and scheduling tasks, Coordination & Communication: organizing and matching up the team, and Testing: improving the activities in the Test phase of the existing software development process. The agile and lean practices included came from [17] [18] [19], and can be summarized as follows:

- Parallelization of the Testing activities with the Design and Programming traditional activities.
- Planning and designing tests before coding.
- Merging the testing team within the development team at early stages.
- Avoiding giving a formal track to the discovery of defects.
- Making a best estimate of activities including the perspective of each member of the developing team.

Table 2. Fragment of the cookbook (left: old activity, right: proposed new activity)

Software Development Process		PAL Activities
A.7 Test Phase		
Previous Activities	PAL Activities	
1. Distribute tasks to team members according to their roles.	Avoid... Test department. Sometimes the test department assigns the testers at the end of the iteration, this is not recommended.	
2. Design the Test Cases.	Avoid... Assuming testing means testing. Testing does not start after coding is completed; specify the system behaviour by writing the test cases.	
3. Check the Test Cases.	Try... Simple testing classifications. The test design should be guided by an expert, concentrating the efforts of the testing in business-critical requirements.	
4. Fix the System Test Cases based on the Verification Report.		
5. Validate System Test Cases.	Try... Test the walls. Sketch the design of the test cases on a whiteboard. This promotes collaboration and team review.	
6. Fix and adapt, if needed, the System Test Cases based on the Validation Report.	Avoid... Separating development and testing. Tracking this kind of development involves merging the test team within the development team, another direct consequence of this action is a better planning and allocation of activities.	
7. Execute tests into the test environment.		
8. Register defects into the Defect Tracking System.	Avoid... Using defect tracking systems during the iteration. If it takes a lot of time and effort, a task on the <u>Sprint Backlog</u> is created.	
9. Perform System Acceptance Testing.		
10. Correct the defects found.	Try... Zero tolerance on open defects. This prevents from spending effort on tracking and prioritizing them, and from delays caused by waiting to fix them.	
11. Verify and close defects.		

The introduction of new practices was guided to accomplish the objectives covered by the previous activities. Each new practice responds to a necessity of changing the way of how to do it, not what to do, that is, the old set of practices is equivalent to the new one in terms of achieving the CMMI requirements and goals, because every new

practice proposed is associated with an old practice, consequently, both of them achieve the same goal and objective. Due to the fact that the old set of practices covered the CMMI requirements, the new practices do too.

Once the practices had been identified and validated, the PAL team started the creation of templates and their customization, being this Step 2. The templates are based on SCRUM products: the *Product Backlog*, the *Burn Down Chart* and the *Sprint Backlog* managed into an excel spreadsheet. It is worth mentioning that the election of a spreadsheet rests in two main factors: the first one is that it eases the customization in the first application, and the most important is that the work team does not have access to the web when they stay at the client's facilities.

The *Sprint Backlog* is the prioritized list of tasks to be completed during the sprint, in this case the template also considered who selected each task and their status. In the same template the hours estimated for each task and their daily progress were recorded. See Figure 2.

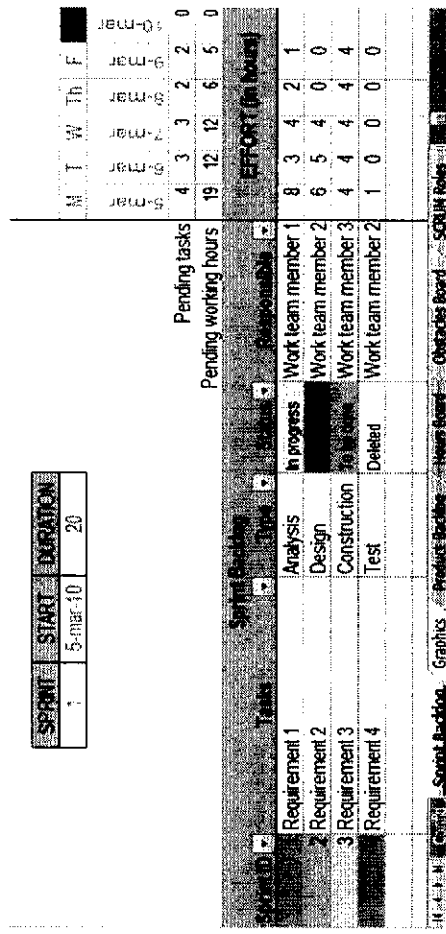


Fig. 2. Fragment of the Sprint Backlog customized template

Taking into account the status and the hours left from each task, the section *Graphics* had three charts, one showing the daily progress of tasks done, another screening the estimated hours left to the end of the sprint, alike *Burn Down Chart*, see Figure 3, and the last one presenting the hours left to the end of the sprint of each work team member.

The *Product Backlog* template was a list of each requirement, with the addition of three values: the name and number of tasks into which the requirement was decomposed, the estimated done date and the sum of estimated hours assigned to it.

Step 3 focused on presenting the principles of agile and lean practices to the work team. Also, a mapping between SCRUM roles and organization's roles, see Table 3, the cookbook and the templates was presented. For that purpose a couple of one-hour sessions with the work team were held. In those sessions the PAL team reinforced the organizational guidelines for tailoring the process.

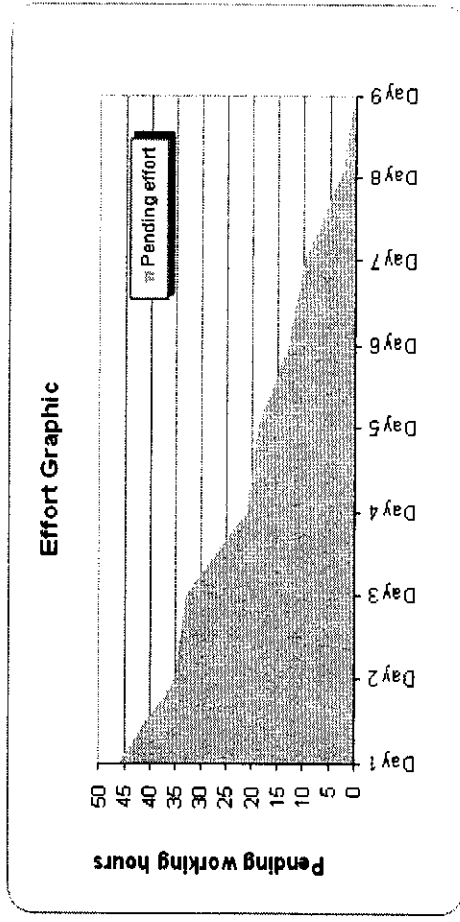


Fig. 3. Fragment of the Burn Down Chart

Table 3. Mapping between roles

Normally played role	SCRUM role
Technical Leader / PAL team	SCRUM Master
Analysts	Team
Designers	
Programmers	
Testers	
Project Manager	Product Owner
Client	

To conclude, Step 4 started when the application of the practices and the data collection began. Taking into account that an often-overlooked difference between agile and traditional engineering processes is the way everyday business is conducted [7], during the first project, PAL1, the work team was helped every day to guide the SCRUM meeting, to introduce the new practices and to collect the data into the templates.

For the second project, PAL2, the work team applied the cookbook by themselves, but the PAL team attended the SCRUM meetings, monitoring and supporting the application of the cookbook.

It is worth mentioning that the effort in hours invested by the two members of the PAL team in the execution of the five steps was 186 hours, starting in the middle of January and finishing about the middle of May, the major part of hours concentrated in the first half of this period.

4.2 Data Collection

The data collection was developed using *Participant Observation*, a systematical and unobtrusive technique defined in [22], through SCRUM meetings and concentrated in the templates. A SCRUM meeting was held every morning, asking the team three

simple questions by the SCRUM Master: What have you done since yesterday's meeting? What are you planning to do today and how many hours will it take you? Do you have any problems preventing you from accomplishing your goal?

For practical reasons the SCRUM Master was played by the Technical leader supported by a member of the PAL team if needed. Having the answers to these questions, the PAL team collected hours, effort and task selection data.

In order to provide a self-contained spreadsheet, the tab *Config* controls specific values like: work team members' name, starting date, status names, phase names and number of sprints.

The tabs *Obstacles Board* and *Hours Board* were added, the first one to describe and manage the obstacles reported by the work team at the SCRUM meetings, such as their type, reporting date, solving date, impact and priority; and the second one, to control the hours assigned to the project by phases and by tasks. The use of those tabs was optional.

4.3 Results

At the end of the PAL projects the goals set initially were fully obtained. To recall, the objectives initially proposed were: increase the customer's satisfaction, improve the work team communication and the bonus of reducing product release time.

Some metrics were collected across the project, but some information had to be obtained by means of direct questions to the involved people. For that purpose, at the end of each PAL projects two Retrospective Meetings took place. The first one was held together with the PAL projects work team, the PAL team, the High Management and the Process Group. At the second one only the presence of the PAL projects work team and the PAL team was required.

These meetings are similar to the Sprint Retrospective meeting in SCRUM, and they had the purpose of reflecting on the past project, identifying what went well during the project, what could be improved, and mainly for our interest, detecting and perceiving the feelings of the team and the client.

Firstly we expand on the communication in the work team. This issue had the most important impact internally; the team members were integrated every day during the SCRUM meetings, managing to achieve direct communication of their needs and concerns and overcoming obstacles together.

There is no metric to show this improvement, but facts like simplicity and early feedback where mentioned constantly in the Review Meetings. In the work team's own words "*The solution to our problems (obstacles) is found at the end of the day, instead of at the end of the week!*".

Besides, not only the communication at work team level was improved, but also the feeling of being heard and supported by the High Management level increased.

Secondly, the constant change of the requirements was managed considerably better. To be more explicit, Figure 4 shows a graphic of pending tasks with variables of working days and tasks number in similar projects, where a sudden appearance of new tasks in the middle of a project is clearly observed.

On the contrary, Figure 5 presents the number of pending tasks in PAL2 project, where the planning and estimation of tasks was enhanced, by taking into account new

practices in order to mitigate, since the beginning of the project, the impact of sudden unexpected tasks.

Thirdly, the reduction of the product release time was achieved, in PAL1 project the product was released 6 working days before planned, and in PAL2 5 days earlier. This fact brings about a considerable reduction of the effort invested and a higher motivation. In addition, according to the quality assurance metrics the PAL projects achieved lower defect densities than the historical metrics.

All together it resulted in the increase of the client's satisfaction, demonstrated by a phone call to the High Management praising the great job done and early signing of the delivery agreement.

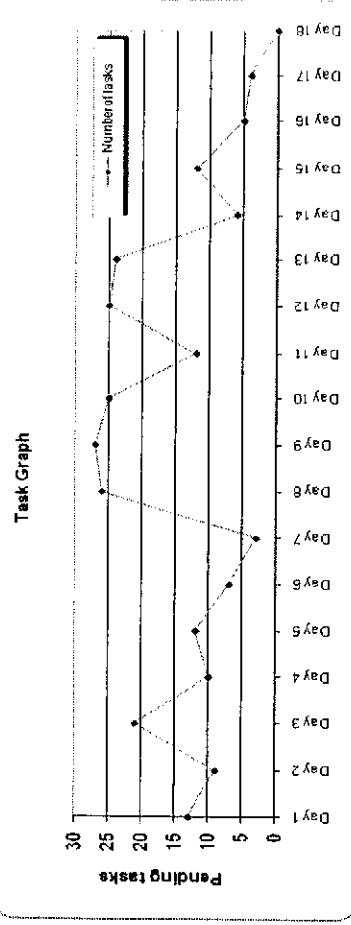


Fig. 4. Task graph of a similar project

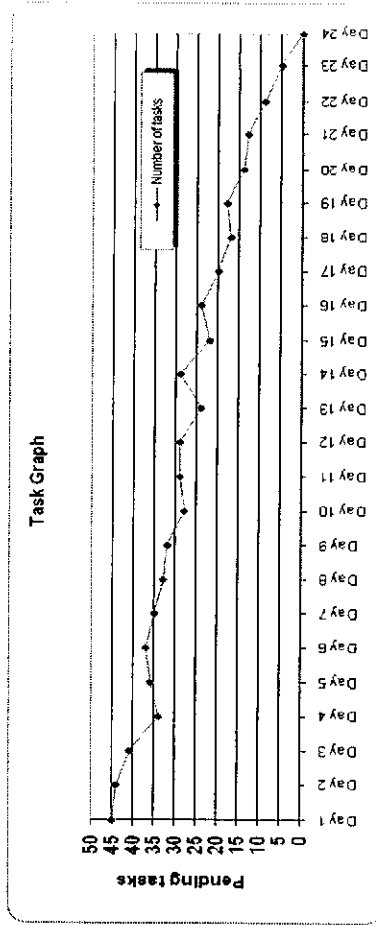


Fig. 5. Task graph of PAL2 project

5 Conclusions and Future Work

Nowadays, many organizations run agile software development projects alongside their traditional ones [20]. For this reason, the experience presented in this work is one more effort intended to clarify and highlight the benefits from using both.

We demonstrated the viability of merging CMMI best practices with agile and lean practices, in order to improve the capacity and maturity of the processes developed by the organization, always trying to preserve stability and helping the integration of agile and lean practices into the traditional process.

An analysis of existing and proposed practices was conducted to identify the project's needs and indispensable process components to be addressed by the agile approach, thus a balanced process was found to substitute the existing one.

As a result of this work, we provided a controlled and guided introduction of agile and lean practices into the organization processes, improving factors like:

- The client's satisfaction increased significantly; we even received a congratulation phone call.
- The work environment of the work team was enhanced, the team's professional and personal development was given a boost, the results obtained were better and the feeling of being heard and taken into account by the High Management increased and was motivating.
- The communication protocols and channels were improved and shortened, which motivated the team to be more collaborative and participating.
- The quality of the final product was improved.
- The estimation and selection of tasks were improved, obtaining better estimations and balanced workloads onto the work team members.

On the other hand, some aspects related to the change resistance were faced, in particular, the mistrust in agile practices caused by the work team members' unawareness. This skepticism was gradually solved by introducing the agile concepts in the workshops and clarifying doubts constantly, but mainly, the team was convinced by the results obtained day by day during the development of the projects.

Another problem was the feeling of "having more work to do", as the introduction of the new practices really required extra time and effort spent by the members involved. So, at the beginning, with the assistance of the PAL team that feeling was controlled, thereafter, the direct benefits obtained from that "more work" convinced the team that the outcome is worth the effort.

Finally through these projects, we highlight and foster the feasibility of integrating of agile and lean practices into traditional practices at an organization certified CMMI level 5. We encouraged success factors involved in improving organizational processes, work environment and product quality.

For future work, we consider the migration of the final customized templates to a tool that minimizes the effort and maximizes the portability. Refining and adapting the cookbook's contents in order to extend its scope to more processes are also considered as useful and beneficial for the organization.

Acknowledgements

This work has been possible thanks to Ultrast's rising spirit of innovation and its desire to improve. Special thanks to the Process Group Leader: Claudia Alquicira Esquivel and the Process Specialist: Hugo García Figueroa. As well as to all the work team members involved in the PAL projects. We also acknowledge the assistance of

the following projects: PEGASO-MAGO (TIN2009-13718-C02-01, FEDER and MEC of Spain) and Agreement Unicauca-UCLM (4982-4901273). Francisco J. Pino acknowledges to the University of Cauca where he works as Assistant Professor.

References

1. Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, S.: CMMI or Agile: Why Not Embrace Both! CMU/SEI-2008-TN-003 (2008)
2. Pino, F., Pedreira, O., Garcia, F., Rodríguez, M., Piatini, M.: Using Scrum to Guide the Execution of Software Process Improvement in Small Organizations. *Journal of Systems and Software* 83(10), 1662–1677 (2010)
3. Dyba, T., Dingsoyr, T.: Empirical studies of agile software development: A systematic review. *Inform. Softw. Technol.* (2008)
4. Cockburn, A., Highsmith, J.: Agile software development, the people factor. *Computer* 34(11), 131–133 (2001)
5. Boehm, B., Turner, R.: *Balancing Agility and Discipline - A Guide for the Perplexed*. Pearson Education, London (2004)
6. Lepmets, M., Nael, M.: Balancing scrum project management process. In: Sillitti, A., Martin, A., Wang, X., Whitworth, E. (eds.) XP 2010. *Lecture Notes in Business Information Processing*, vol. 48, pp. 391–392. Springer, Heidelberg (2010)
7. Boehm, B., Turner, R.: Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software*, 30–39 (2005)
8. Poppendieck, M., Poppendieck, T.: *Lean Software Development: An Agile Toolkit*. Addison-Wesley, Reading (2003)
9. Lean Software Institute, <http://www.leansoftwareinstitute.com/>
10. Poppendieck, M., Poppendieck, T.: *Implementing Lean Software Development: From Concept to Cash*. Addison-Wesley, Reading (2006)
11. Appelo, J.: Complexity vs. Lean, the Big Showdown. In: *Lean Software and Systems Conference 2010*, Helsinki (2010)
12. Ultrast, <http://www.ultrast.com.mx/>
13. Schwaber, K., Beedle, M.: *Agile Software Development with SCRUM*. Prentice Hall, Englewood Cliffs (2001)
14. Beck, K.: *Test-Driven Development by Example*. Addison-Wesley, Reading (2003)
15. Dávila, M., Oktaba, H.: *Especialización de MoProSoft basada en el Método Ágil Scrum (MPS-Scrum)*. In: SIS 2009 (2009) (in press)
16. Jakobsen, C., Johnson, K.: *Mature Agile with a twist of CMMI*. In: *Agile 2008 Conference*, pp. 212–217 (2008)
17. Larman, C., Vodde, B.: *Practices for Scaling Lean & Agile Development*. Addison-Wesley, Reading (2010)
18. Larman, C.: *Agile & Iterative Development: A Manager's Guide*. Addison-Wesley, Reading (2004)
19. Coplien, J., Bjornvig, G.: *Lean Architecture*. Wiley, Chichester (2000)
20. Salo, O., Abrahamsson, P.: *Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study*. In: *International Symposium on Empirical Software Engineering 2005*, pp. 193–202 (2005)
21. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Dordrecht (2000)
22. Shull, F., Singer, J., Sjöberg, D.: *Guide to Advanced Empirical Software Engineering*. Springer, Heidelberg (2010)