

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

#### Publisher

UPGRADE is published on behalf of CEPIs (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by NOVÁTICA <http://www.ati.es/novatica/>, journal of the Spanish CEPIs society ATI (Asociación de Técnicos de Informática <http://www.ati.es/>).

UPGRADE is also published in Spanish (full issue printed, some articles online) by NOVÁTICA, and in Italian (abstracts and some articles online) by the Italian CEPIs society ALSI <http://www.alsi.it> and the Italian IT portal Tecnoteca <http://www.tecnoteca.it/>.

UPGRADE was created in October 2000 by CEPIs and was first published by NOVÁTICA and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>).

#### Editorial Team

Chief Editor: Rafael Fernández Calvo, Spain [rfoalvo@ati.es](mailto:rfoalvo@ati.es)  
Associate Editors:

• François Louis Nicolet, Switzerland, [nicolet@acm.org](mailto:nicolet@acm.org)  
• Roberto Carniel, Italy, [carniel@dgt.uniud.it](mailto:carniel@dgt.uniud.it)

#### Editorial Board

Prof. Wolffried Stucky, CEPIs President  
Fernando Piera Gómez and  
Rafael Fernández Calvo, ATI (Spain)  
François Louis Nicolet, SI (Switzerland)  
Roberto Carniel, ALSI – Tecnoteca (Italy)

**English Editors:** Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green, Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson.

Cover page designed by Antonio Crespo Foix, © ATI 2003

Layout: Pascale Schürmann

E-mail addresses for editorial correspondence:  
[nicolet@acm.org](mailto:nicolet@acm.org) and [rfoalvo@ati.es](mailto:rfoalvo@ati.es)

E-mail address for advertising correspondence:  
[novatica@ati.es](mailto:novatica@ati.es)

Upgrade Newsletter available at

<http://www.upgrade-cepis.org/pages/editinfo.html#newsletter>

#### Copyright

© NOVÁTICA 2003. All rights reserved. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, write to the editors.

The opinions expressed by the authors are their exclusive responsibility.

ISSN 1684-5285

Next issue (Oct. 2003):  
“e-Learning – Borderless  
Education”

## Software Engineering – State of an Art

Guest Editor: Luis Fernández-Sanz

### Joint issue with NOVÁTICA

#### 2 Presentation: Software Engineering. A Dream Coming True? – Luis Fernández-Sanz

*The guest editor presents the issue, that focuses on a really broad field like Software Engineering (SE) which has been driving the evolution of software development since the late sixties of the past century. The papers cover different areas of interest related to the application of engineering principles to software development and maintenance. As usual, a list of useful references is also included for those interested in knowing more about this subject.*

#### 5 Software Project Management. Adding Stakeholder Metrics to Agile Projects – Tom Gilb

*In this article the author offers an analysis of the implications of the new agile methods in the field of software development.*

#### 10 Model-Driven Development and UML 2.0. The End of Programming as We Know It? – Morgan Björkander

*This paper focuses on the idea of a truly model-driven software and analyses the influence that the new version of UML (Unified Modeling Language) is having on this process.*

#### 15 Component-Based Software Engineering – Alejandra Cechich and Mario Piattini-Velthuis

*This paper studies the important role that components play in the field of Software Engineering.*

#### 21 An Overview of Software Quality – Margaret Ross

*The author reviews important issues concerning quality in software development and also deals with the issues of users with disabilities and the influence of legislation regulating this aspect.*

#### 26 Lessons Learned in Software Process Improvement – José-Antonio Calvo-Manzano Villalón, Gonzalo Cuevas-Agustín, Tomás San Feliu-Gilabert, Antonio de Amescua-Seco, M<sup>a</sup> Magdalena Arcilla-Cobián, and José-Antonio Cerrada-Somolinos

*This paper describes the lessons learned by SOMEPRO, a Software Engineering R & D group in the Universidad Politécnica de Madrid, in more than ten software process improvement projects.*

#### 30 A New Method for Simultaneous Application of ISO/IEC 15504 and ISO 9001:2000 in Software SME's – Antònia Mas-Pichaco and Esperança Amengual-Alcover

*The authors offer their view, based on practical experiences, of the always thorny problem of applying best software development practices to organizations where resources are especially limited.*

#### 37 Empirically-based Software Engineering – Martin Shepperd

*This paper presents an overview of empirical Software Engineering and its implications for practitioners and researchers in four areas (object-orientation, inspections, formal specification and project failure factors.)*

#### 42 Software Engineering Professionalism – Luis Fernández-Sanz and María-José García-García

*The aim of this paper is to provide a brief overview of what goes into making up our true perception of software engineers as specialised professionals within the field of Information Technologies.*

#### 47 Searching for the Holy Grail of Software Engineering – Robert L. Glass

*In this article, the author defends eclecticism in development methods and the contribution that Software Engineering should make in this respect whenever the nature of a project demands flexible methods in order to be successful.*

#### 49 Free Software Engineering: A Field to Explore – Jesús M. González-Barahona and Gregorio Robles

*This article analyses the existing points of contact between Software Engineering and the development of free software, and puts forward a few future lines of research in this respect.*

# Lessons Learned in Software Process Improvement

*José-Antonio Calvo-Manzano Villalón, Gonzalo Cuevas-Agustín, Tomás San Feliu-Gilabert, Antonio de Amescua-Seco, M<sup>a</sup> Magdalena Arcilla-Cobián, and José-Antonio Cerrada-Somolinos*

*Nowadays, Software Process Improvement (SPI) is thriving in the software industry. This paper identifies the phases of a software process improvement project and describes the lessons learned by SOMEPRO, a Software Engineering R & D group in the Universidad Politécnica de Madrid, in more than ten software process improvement projects in Spain, detailing the aspects an organization has to take into account in order to carry out this kind of projects.*

**Keywords:** capability maturity level, CMM, software appraisal, software process assessment, software process improvement.

## 1 Introduction

The demand for better services and functionality of software products is continual. The need for products to be converted into new ones in which the architecture is changed (affecting the product complexity), and the necessity for rapid development are mandatory requirements. In this context, traditional software production problems remain: system delivery delays, high costs, low product quality, and difficult maintenance.

Expectations in technology, with regard to the use of new tools and methods, have not delivered the desired response during the last decade. Nowadays, a movement based on the software process improvement (SPI) principle, “*the quality of a software product depends on the quality of processes followed to elaborate it*”, is being established with solid, positive, and lasting results.

SOMEPRO is a Spanish group of professors whose main purpose is research and development in the Software Engineering area, especially in software process improvement. In 1993, SOMEPRO started the first SPI project carried out in Spain [1] in collaboration with the North American organization ISPI (Institute for Software Process Improvement). Since then, SOMEPRO has participated in projects funded by the European Commission [2][3][4] and private enterprises.

SOMEPRO normally participates in Software Process Improvement (SPI) projects with other organizations. The main activities in which SOMEPRO participates are:

- Software Process Assessment.
- Design of Software Process in accordance with reference models such as CMM (Capability Maturity Model), ISO 15504, ISO 9001.
- Technology Transfer in different areas of software engineering.

SOMEPRO has participated in software process improvement projects in small and medium-sized enterprises [5] (from 50 to 500 employees) whose activities are mainly banking, telecommunications, engineering and administration. All of

them develop software for their own organization or for their head office. Their maturity level was under 2 when they started the improvement. After implementing process improvements

*José-Antonio Calvo-Manzano Villalón* has a Ph. D. in Computer Science. He is assistant professor in the Computer Science Faculty of the Universidad Politécnica de Madrid, Spain. He is teaching in the area of Software Engineering. He is author of several international papers related to the software engineering area. He is author of books in the software process improvement and software engineering topics. He has collaborated in several European projects oriented to software risk management, software process improvement in requirements management. He has participated in several projects with enterprises related to manufacturing process improvement and also software process improvement. <jacalvo@fi.upm.es>

*Gonzalo Cuevas-Agustín* has a Ph. D. in Telecommunications and a MS in Computer Science. He is actually full professor in the Universidad Politécnica de Madrid, Spain, on Software Engineering. He has occupied senior management positions during more than 30 years in the university and industry always in the environment of information and communication technologies. Founder and Director of Master in Software Engineering in association with Carnegie Mellon University in the Computer Science Faculty of Polytechnic University of Madrid. He is reviewer of the European Commission and consultant in software process improvement. Current research interest: models and methods for software process improvement. Senior member of IEEE and member of ACM, author of numerous publications and books on software engineering and speaker at numerous conferences, seminars and symposia in Spain and abroad. <gcuevas@fi.upm.es>

*Tomás San Feliu-Gilabert* has a Ph. D. in Computer Science. He is assistant professor in the Computer Science Faculty at the Universidad Politécnica de Madrid, Spain. He teaches Software Engineering in this Faculty. He is author of several international papers related to the software engineering area. He is author of books in the software process improvement and software engineering topics. He has collaborated in several European projects oriented to software risk management, software process improvement in requirements management. <tsanfe@fi.upm.es>

*Continued on next page*

over an average period of 24 months, some of them achieved level 2.

## 2 Best Practices Reference Models

Because software is fundamental in all sectors, research centres or institutions began to collect practices from organizations which have reached outstanding results in their software life cycle process. These practices are called best practices and are organized into processes. A reference model is a set of processes that guides organizations to determine their capability level and to improve their software process.

The most known and extended model in the software community is the Capability Maturity Model (CMM) for Software [6][7]. CMM was developed by the SEI (Software Engineering Institute) of the Carnegie Mellon University (USA) and was sponsored by the Department of Defense (DoD) of the United States. Another model being used is the ISO 15504 “Information Technology – Software process assessment – Part 2 A reference model for processes and process capability” [8]. Recently, these two models have been integrated into the Capability Maturity Model Integration (CMMI) Continuous and Staged Representations [9][10].

## 3 The Software Process Improvement Cycle

Process improvement follows an incremental and cyclical approach. The software process improvement cycle proposed by the ISPI is represented in Figure 1 [11].

- Commitment to SPI. The objective of this stage is to obtain senior management commitment with regard to people, time and other resources needed in the software process improvement project.
- Software process appraisal. The objective of this stage is to determine the organisation's maturity level, and strengths and weakness of the processes assessed with respect to CMM, in order to select processes to be improved.
- Infrastructure and plans for SPI. One objective of this stage is to define the infrastructure needed to carry out improvements in the previously selected processes. Another objective is to create an action plan for selected processes in order to define and implement improvements.

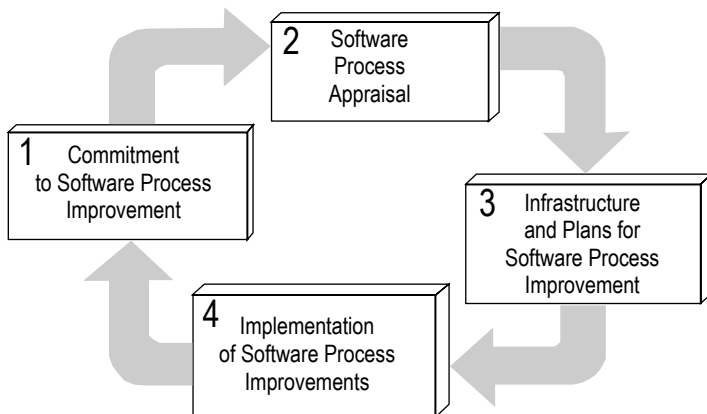


Figure 1: ISPI's Software Process Improvement Model

*Antonio de Amescua-Seco* has a Ph. D. in Computer Science. He is full Professor in the Computer Science Department of the Universidad Carlos III de Madrid, Spain. His research areas are software engineering methodologies and software process improvement. He has also worked in Polytechnic University of Madrid and Iberia Airlines Company. He was the research project leader for the development of the Information System Development Methodology for the Spanish Administration and has participated in projects sponsored by the European Union.  
<amescua@inf.uc3m.es>

*Magdalena Arcilla-Cobián* has a BSc. in Computer Science, Computer Science Faculty, Universidad Politécnica de Madrid, Spain. She got her Advanced Studies Diploma in the Languages & Computer Science Department of the UNED (Spanish Open University) She is assistant professor in the Languages & Computer Science Department of the UNED (Universidad Nacional de Educación a Distancia – Spanish Open University). She is teaching in the area of Software Engineering. She has also worked in the Computer Science Department of the Universidad Carlos III de Madrid, Spain, and she has 8 years of experience as software developer for CustomWare Company. She is author of several international papers related to the software engineering area. She is author of a book in the software engineering topics.  
<marcilla@lsi.uned.es>

*José-Antonio Cerrada-Somolinos* has a Ph. D. in Industrial Engineering at the Universidad Politécnica de Madrid, Spain. He is actually full professor in the UNED (Universidad Nacional de Educación a Distancia – Spanish Open University) on Software Engineering. His research areas are software engineering methodologies and software process improvement. He is consultant in software process improvement. Author of numerous international papers and several books in the software process improvement and software engineering topics. Current research interest is a new information system design methodology called ERA (Entity Relation Automata). Member of IEEE and IFAC.  
<jcerrada@lsi.uned.es>

- Implementation of SPIs. The objective of this stage is to implement each action plan generated, define the new process, implement it in a pilot project and, finally, institutionalize it while refining it when feedback is obtained.

## 4 Lessons Learned

In this section, some general lessons learned as well as lessons relating to the different stages from the previous improvement model are presented. SOMEPRO has been working with the ISPI's process improvement model and the CMM as best practices model.

### 4.1 General Lessons Learned from the Improvement Process

The acceptance of a process model of reference greatly simplifies the adoption of generic practices in software processes. However, statistics show that a high percentage of implementations are not successful. Frequently, the reasons for this failure are not technical; instead they are due to inadequate change management which every improvement entails. A successful SPI implementation

requires the accomplishment of the following principles included in the Total Quality practice:

- *Senior management commitment* in process improvement must be embedded in all decisions to be made throughout the improvement project, and these decisions must be communicated to all personnel involved. Frequently, the most compromising decision is to assign resources to software process improvement when these resources are shared with the productive tasks.
- Improvement is *participative* and is developed *within* the organization. The personnel who make up the productive teams are those who develop improvement actions. External personnel can be in charge of training but the participation of those involved in the processes is fundamental.
- Improvement usually requires a great deal of *training* of the improvement team, not only in the process model of reference (CMM) and new techniques and tools but also, in most cases, in the principles of change management.
- Improvement is *instructive*. The objectives, development and plans are communicated openly and periodically to those involved in improvement.
- Improvement must be treated as a *collaborative* effort between the different personnel involved in the improvement process. The improvement promoter and quality assurance roles have evolved from audit and control toward *change facilitators*, supporting at all times those who are experimenting with new processes whenever difficulties arise.

It has been proved that the accomplishment of these principles result in sound and lasting implementation throughout the organization, with smooth transitions to new ways of working, and contribute effectively to personnel satisfaction.

#### 4.2 Lessons Learned in the ISPI's Model Stages

To better understand the lessons learned, the principal activities SOMEPRO applied in each stage of the improvement model are indicated.

##### 4.2.1 Commitment to SPI

Activities in this stage are:

1. Selecting personnel involved in the appraisal – appraisal team, Software Engineering Process Group (SEPG), project managers and other personnel.
2. Preparing the environment, including logistics.
3. Selecting the improvement processes to be assessed (optional).
4. Training the appraisal team and the SEPG in the process model of reference (CMM), team working and change management, and how to conduct an appraisal.

Lessons learned in this stage:

- Personnel's initial resistance to change.
- Selected personnel's interest and commitment to improvement to achieve low resistance to change.
- Personnel should be selected to form part of the SEPG.
- Senior management must be explicitly committed; for instance, informing personnel about the improvement policy.

- The improvement project, like any other project, must have a project manager.
- Some practices of maturity level 3, such as testing or peer reviews, may have more priority than some processes of maturity level 2 due to business needs.

##### 4.2.2 Software Process Appraisal

Activities in this stage are:

1. CMM training and filling in the questionnaire.
2. Preliminary analysis of the answers.
3. Selection of the processes to be improved.
4. Analysis of inconsistencies given in the answers by interviewing those involved in the selected processes.
5. Presentation of findings and recommendations.
6. Prioritization of processes to be improved and guidance for action plans.

Lessons learned in this stage:

- SEPG personnel must take part in the appraisal process to acquire technology transfer.
- Appraisal has to be participative. The personnel of the organization receive the necessary training and conduct their own appraisal.
- Project managers who answer the questionnaire must have sufficient knowledge of the company in general, and the routine way of managing the projects.
- A wider variety of answers than *yes/no* needs to be established. For example, we use five types of answers – *always, usually, often, rarely and never* – in order to be better informed of the state of the process.
- The questions in every 'key process area' must be answered on completion of the explanation of the key process area, and not after the whole CMM training, in order for answers to be truthful and appropriate to what the questionnaire defines.
- Answers to the questions usually will focus on the way the organization works in general and on what the project managers know about the organization rather than on the specific project.
- No more than three processes, based on the information from the questionnaire, coverage of the key process area and the priorities of the business, should be selected for improvement.
- Experienced personnel with prestige in the organization should be selected in order to overcome change resistance later. It is also advisable to inform the interviewers of the results obtained from the interviews and to allow them to give feedback.
- Interviews should be prepared in great detail, including all the problems arising from the answers given in the questionnaire.
- As a result of the appraisal, a draft of the action plan is included in order for senior management to continue with the improvement project and to be aware of the activities and tasks needed to be carried out in the following stages.
- Short term results are defined as part of the action plan guidance in order to encourage the organization to continue with the process improvement project.

- Senior management should be given the presentation before the rest of the personnel involved in the process improvement to make sure that information that senior management considers to be inappropriate is not provided.

#### 4.2.3 Infrastructure and Plans for SPI

Activities in this stage are:

1. Defining and establishing the organizational infrastructure for improvement.
2. Planning the improvement process, including activities like defining the new process, training the working groups and the pilot project team, and selecting the pilot project for each improvement selected.

Lessons learned in this stage:

- The improvement infrastructure must be established for the smooth running of the project.
- The organization must treat the process improvement as priority so that they do not take their experienced and prestigious personnel off the project for other activities.
- SQA (Software Quality Assurance) and SEPG (Software Engineering Process Group) personnel are included in the same organizational unit at the beginning of the process improvement due to the lack of quality culture in the organizations.
- Personnel with enough experience and prestige in the organization should be chosen to compose the working groups of the processes to be improved so that the processes defined are accepted by the rest of the organization.
- The leader of the working groups should be a member of the SEPG so that further technology transfer will be easier.
- The pilot project should only last between 4 and 6 months so that improvement results are visible within a short period of time. It should not be complex, but have added value for the organization. In addition, the project manager should not oppose the improvement process.

#### 4.2.4 Implementation of SPI

Activities in this stage are:

1. Defining the new process.
2. Implementing the process in the pilot project and refining it (as a result of the lessons learned).
3. Institutionalizing the new process.

Lessons learned in this stage:

- The working group personnel must be trained in the process they are going to be involved in.
- The pilot team and the rest of staff must be trained in the new processes.
- The Working Group and the SEPG must support the pilot team. In turn, the SEPG must support the entire organization to institutionalize the process.
- For each new process to be implemented, the metrics must be defined in order to determine the efficacy of the new process and its impact on the business.
- The change must be introduced gradually, first with a small group and then extending it until it is institutionalized.

## 5 Conclusions

Surviving in a more and more competitive software world requires more than contracting software engineers with knowledge and purchasing new development tools. It is also necessary to use effective software processes so that engineers can routinely use the best technical and management practices to succeed in their projects. More and more organizations consider SPI as a way to improve quality, productivity and effort estimations, software acquisition and maintenance.

The lessons learned constitute a set of requirements to take into account and/or risks to mitigate in order to avoid SPI project failure.

## References

- [1] IBERIA, LAE. SPIE: Software Process Improvement and Experimentation, ESSI Project No 10344, February 1994.
- [2] EROSKI SCOOP. SYMQUAD: System Requirements and quality assurance in a distribution company, ESSI Project No 23947, March 1997
- [3] LANTIK S.A. LANHOBK: Client Relationships and Requirement Management Improvement, ESSI Project No 24155, March 1997
- [4] G.O.C. HOSPUR: User Requirements and Reuse in Health Management Systems, ESSI Project No 24181, March 1997
- [5] J. A. Calvo-Manzano, G. Cuevas, T. San Feliu, A. Amescua, L. García and M. Pérez. Experiences in the Application of Software Process Improvement in SMEs, *Software Quality Journal*, 10, pp. 261–273, 2002
- [6] M. C. Paulk, S. M. Garcia, M. B. Chrissis and M. Bush. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, Technical Report, Software Engineering Institute, Carnegie Mellon University, 1993.
- [7] M. C. Paulk, S. M. Garcia, M. B. Chrissis and M. Bush. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-25, Technical Report, Software Engineering Institute, Carnegie Mellon University, 1993.
- [8] International Standard Organization and International Electro-technical Commission. ISO/IEC Software Process Assessment Information Technology – Version 3.3, ISO/IEC 1998, JTC1/SC7/WG10, TR 15504: 1998.
- [9] CMMI Product Team. Capability Maturity Model Integration (CMMISM), Version 1.1; CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing, (CMMI-SE/SW/IPPD/SS, V1.1), Continuous Representation, CMU/SEI-2002-TR-011, Carnegie Mellon, Software Engineering Institute, Pittsburgh, 2002.
- [10] CMMI Product Team. Capability Maturity Model Integration (CMMISM), Version 1.1; CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing, (CMMI-SE/SW/IPPD/SS, V1.1), Staged Representation, CMU/SEI-2002-TR-011, Carnegie Mellon, Software Engineering Institute, Pittsburgh, 2002.
- [11] Institute for Software Process Improvement. Process Improvement Model, Seminario Software Process Assessment Training, Institute for Software Process Improvement, Iberia, Madrid, March 1994.