

OWPL Micro Assessment

Naji Habra
University of Namur
21 rue Grandgagnage
Namur, Belgium
nha@info.fundp.ac.be

Alain Renault, Simon
Alexandre, Miguel Lopez
CETIC
8 rue Clement Ader
Gosselies, Belgium
{are,sa,malm}@cetic.be

ABSTRACT

This paper describes an experience with a Software Process Assessment (SPA) tool particularly adapted to small structures (e.g. small and medium enterprises in software businesses, small software teams in bigger businesses, small software teams in public organizations). A characterization of Software Process in-the-small is first made on basis of a deep analysis of software processes actually followed by a number of small and medium enterprises (SME's). The approach proposes a gradual Software Process Assessment framework that allows SME's to start Software Process Improvement (SPI) in a very targeted manner, to quickly progress within a limited budget and, eventually, to reach an acceptable maturity level. The experience has shown the relevance of this method.

Keywords

Software Process, Software Process Assessment, Software Process Improvement, Software Quality, Software Process in-the-small, OWPL, Software Process Improvement in-the-small

1. INTRODUCTION

Most of the big companies are aware of quality matters and invest time and money in improving their software products. Some of them have created their own reference models but most are using standard models developed by international organizations[7]. The models now available can be applied to almost any field of the industry. However, those models are rather unusable for very small businesses as they are too much complicated and too expensive to implement. A solution to that problem was to be found and a software improvement approach appropriate for the very small development structures has been developed.

The approach underlying construction of this model is based on a hypothetico-deductive reasoning. The scope of the model is derived from strategic objectives and high level

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICSE 2002 Orlando, Florida USA

Copyright 2001 ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

goals. The targets of the model were then deduced from those objectives and from previous experience with small companies. To build an operational approach a set of six hypothesis concerning the targets and the framework have been made. In other words, the relevance of the collected information depends on the verification of the hypothesis. Finally, several means are implemented to enable the achievement of the above goals.

2. TARGET AND GOALS

2.1 Goals

2.1.1 Strategic objective

The strategic objective underlying this study is the improvement of the global maturity level of software practices in SME¹. The first step to achieve this is, at company level, to allow software VSE² to implement SPI.

Those SME and VSE are not only software companies developing packaged or bespoke software for sale, but also small software departments in much larger organizations in any industry sector.

2.1.2 High Level Goals

Four high level goals support this strategic objective :

- Sensitize the organizations to quality issues
- Make a high level inventory of the organization
- Prioritize the steps of the improvement process
- Allow organizations to achieve Software Process Assessment with very few budget and resources involved

2.1.3 Operational Goals

Furthermore, operational goals are specified to find out means to realize the high-level goals :

- Sensitize the sponsor of the targeted organization to the Software Process Improvement
- Minimize the cost and effort necessary to implement Software Process Assessment
- Provide a lite Software Process Assessment tool

¹Small and Medium Enterprise : organizations employing fewer than 50 software engineers

²Very Small Enterprise : organizations employing fewer than 10 software engineers

2.2 Targets

Previous experience[1] with SME and VSE, and other research in this particular sector[2] conducted to focus the scope of the study and to restrict the target to three kinds of software organizations. This categorization is based on three criteria : the size, the maturity level and the technical skills of the software organization. These categories can overlap.

- Organizations with low Software Engineering maturity level
- Organizations with small software development teams
- Organizations with unsuitable technical skill level

3. APPROACH KEY FEATURES

3.1 Hypothesis

At the heart of constructing the framework there are six fundamentals assumptions or hypothesis :

- VSE have a low maturity level in terms of Software Engineering
- Software Engineering concepts and methods are unusable and inaccessible for the targeted VSE
- One single person has a suitable visibility of the software organization
- The visibility of the organization is significant in terms of collected information
- Software development is organized in teams (team work). Any isolated person would be considered as a team
- There is a need for a tailored tool fitted to size and maturity level to proceed with Software Process Assessment and Improvement

3.2 Gradual Approach

Those hypothesis, combined with previously defined goals and targets led to the following approach.

First, a Micro-Assessment Framework is used to collect information about the current software practices in the organization. Those information are analyzed in order to produce a report that summarizes the current state of software practices together with some recommendations to improve the quality of these practices. At the same time, the managers are sensitized to the importance of software quality aspects.

The collected information can then be used as a starting point to determine the goals of a more accurate evaluation according to the OWPL model which has been developed on the same bases though being much more complete and addressing a wider targets[4].

A SPICE[3] or a CMM-based[6] evaluation can eventually be done in bigger organizations, if this appears appropriate.

This approach is not linear. Organizations loop at the level offering the most appropriate method and tool according to their size and software maturity level.

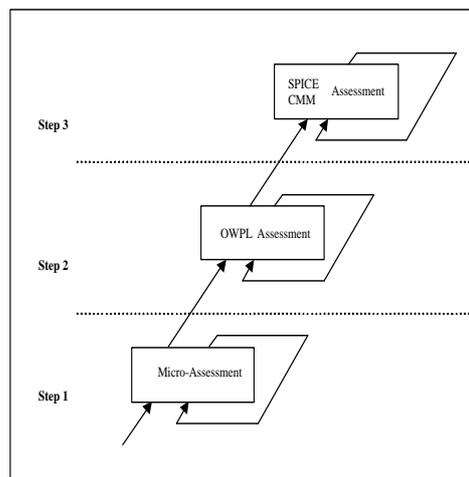


Figure 1: The Gradual Approach

3.3 Means

Five necessary means are provided to implement high level and operational goals :

- Self benchmarking
- Structured framework for collecting data
- Phone based interview (30 to 45 min)
- Simplified terminology
- Track record

4. THE MICRO ASSESSMENT FRAMEWORK

4.1 Key axes

The Micro-Assessment framework covers six key axes selected as the most pertinent and the most prior according to the defined targeted organizations. Those key axes are :

- Quality assurance
- Customers management
- Subcontractors management
- Project management
- Product management
- Training and human resources management

4.2 Questionnaire

Evaluations are performed by quality practitioners, the interviewed person should be the one who has the software quality in his charge in the evaluated organization ; this corresponds usually to one of the executive staff members or to the quality engineer, if this function exists.

The questionnaire includes a few dozens of questions covering the axes above. Questions are open, and each of them is associated with one or more sub-questions allowing the

interviewer, if need be, to adjust and refine the information he gets.

Answers are interpreted according to a fixed grid. Two types of questions can be distinguished. On the one hand, questions that concern essential practices related to the general organization are rated on a linear scale according to the quality of the practice assessed. On the other hand, questions that concern the software practices are rated in a double-entry grid according to the quality of the practice and to its effective implementation in the evaluated organization (only for some critical projects vs. for all projects).

4.3 Example

Here is an example of a question from the phone-based interview :

Do you express formally your customer's requirements ?

*Do you produce a Software Requirement Document ?
Do you have a document with the features required by the customers ?
Does the customer review the above document for validation ?*

Here is the evaluation grid of the above question :

	On some projects	On the whole
Yes, internally		
Yes, with customer approval		

4.4 Report

The result of the micro-evaluation is drawn up in a report of a dozen of pages. A typical report first presents briefly the approach, then it develops the results of the questionnaire and summarizes them according to the six axes. Then it analyzes those results according the situation of the evaluated organization (the age, the history, the declared goals,..) and finally gives some recommendations to help the assessed unit to improve.

5. EXPERIENCES

The micro-assessment has been experimented on a sample of thirty representative organizations (IT small companies, IT services in other businesses, public administrations with an IT department)[5].

It has not been possible to define a general trend that would qualify the soft maturity level of those organizations. Two possible reasons can explain this observation : The size of the sample is too small or actually, there is no trend to be observed.

Some small organizations confirm a weakness in the software development process, as expressed in the first hypothesis (see section 2.1). This corresponds to a development without any well-distinguished phases or even any notion of lifecycle.

But, in bigger organizations the same weakness (lack of lifecycle) has been observed. This fact lead to invalidate the first hypothesis which binds the maturity level to the size of the software organization.

Moreover some very small organizations have pretty good software practices. This led to invalidate the very first hypothesis and to open new research issues.

6. CONCLUSION

The micro-assessment leads its quality practitioner to contact one single person, restricting the evaluation to his/her own point of view. This characteristic can be identified as a limitation of the approach. This is the reason why the contacted person must have a suitable visibility of the organization (see the third hypothesis section 2.1). This allows to get a global but relevant view of the software organization.

One of the goals was to provide to VSE and SME with a low cost evaluation tool. This is really a success as the evaluation process takes less than 45 minutes on company's side.

The global approach seems to have met its objectives, as far as customers feedback can be considered as significant indicator.

7. REFERENCES

- [1] The spire handbook. better, faster, cheaper software development in small organisations. Technical report, The European Community, 1998.
- [2] CITA. Utilisation des systemes d'information inter-organisationnels [sio] par les pme belges. research report of sio. Technical report, University of Namur, 1997.
- [3] K. Emam, J. El, J. Drouin, and W. Melo. Spice: The theory and practice of software process improvement and capability determination, 1998.
- [4] N. Habra, E. Niyitugabira, A.-C. Lamblin, and A. Renault. Software process improvement in small organizations using gradual evaluation schema. In *PROFES'99. Proceedings of the International Conference Product Focused Software Process Improvement*, Oulu, Finland, 1999.
- [5] N. Habra, E. Niyitugabira, A.-C. Lamblin, and A. Renault. Software process improvement in small structures : First results of a micro-assessment framework. In *Proceedings of the European Conference on Software Process Improvement*, Barcelona, Spain, 1999.
- [6] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber. The capability maturity model for software, version 1.1. Report CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.
- [7] S. Zahran. Software process improvement, 1998.