



***OWPL***  
Software Process Improvement  
for VSE, SME and low maturity enterprises

Version 1.2.2



University of Namur,  
Computer Science Dpt



cetic  
Your connection to  
ICT research

**CETIC**  
Centre of Excellence in Information and  
Communication Technologies

Translation in English by  
CETIC asbl

OWPL-Model\_V1-2-2-EN-V22.doc, version of 07/03/2006

This document is the intellectual propriety of the University of Namur Software Quality Lab and CETIC. However, information contained in this document is free of use. The distribution of all or parts of this document is authorized for non commercial use as long as the following legal notice is mentioned:

© **FUNDP-CETIC 2000 – 2005 Software Quality Lab.**

*Commercial use of this document is strictly forbidden.*

This document is distributed in order to enhance exchange of technical and scientific information.

THIS FUNDP AND CETIC MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. FUNDP AND CETIC MAKE NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL.

The Software Quality Laboratory and CETIC asbl can in no manner be considered responsible or liable for errors, mistakes or omissions that the document could contain.

In a litigation case, only Belgian laws will apply and only Belgian courts of law will be considered competent to rule on such a case.

Further information about this document or associated research subjects can be obtained by contacting CETIC asbl.

**<http://www.cetic.be/>**

## List of Authors/Contributors

*Alain RENAULT*

*Naji HABRA*

*Simon ALEXANDRE*

*Jean-Christophe DEPREZ*

*Eustache NIYITUGABIRA*

*Anne-Catherine LAMBLIN*

# Table of contents

<b>1.</b>	<b><i>Introduction</i></b>	<b>5</b>
<b>2.</b>	<b><i>Conceiving the OWPL model</i></b>	<b>6</b>
2.1.	Tailoring constraints	6
2.2.	Structuring the OWPL model	7
2.3.	Structure of the OWPL model	8
2.3.1	Process	9
2.3.2	Practices	9
2.3.3	Success factors	9
<b>3.</b>	<b><i>Compatibility of OWPL with ISO/IEC12207</i></b>	<b>10</b>
<b>4.</b>	<b><i>Detailed presentation of OWPL</i></b>	<b>11</b>
4.1.	Process	11
4.1.1	Requirements Management - REQU	12
4.1.2	Documentation - DOCU	13
4.1.3	Project Planning - PLAN	14
4.1.4	Project Tracking and Oversight - TROV	15
4.1.5	Development - DVLP	16
4.1.6	Testing - TEST	17
4.1.7	Configuration Management - CONF	18
4.1.8	Subcontract Management - SUMA	19
4.1.9	Quality Assurance - QUAL	20
4.1.10	Experience capitalization - CPTL	21
4.2.	Practices	22
4.2.1	Requirements management - REQU	24
4.2.2	Documentation - DOCS	25
4.2.3	Projet Planning- PLAN	27
4.2.4	Project Tracking and Oversight - TROV	29
4.2.5	Development - DVLP	31
4.2.6	Tests - TEST	33
4.2.7	Configuration management- CONF	35
4.2.8	Subcontract management - SUMA	36
4.2.9	Quality Assurance - QUAL	38
4.2.10	Experience Capitalization - CPTL	40
4.3.	Success factors	41
4.3.1	Organization	42
4.3.2	Management	44
4.3.3	Human resources	45
4.3.4	Technology	45
<b>5.</b>	<b><i>OWPL software practices assessment</i></b>	<b>46</b>
5.1.	OWPL assessment method	46
5.2.	Evaluation of Practices	46
5.3.	Success factors evaluation	46
5.4.	Results interpretation	46
5.5.	Evaluation questionnaire	46
<b>6.</b>	<b><i>Bibliography</i></b>	<b>47</b>
<b>7.</b>	<b><i>Annex I: Mapping OWPL – ISO/IEC12207:1995 –AMD1 &amp; AMD2</i></b>	<b>49</b>

# 1. Introduction

The objectives of the *OWPL* project are to define a model for software process assessment and improvement that fits small companies and to help them improve their software practices. Evolution and improvement of practices are considered more important than the actual assessment.

Among other things, the *OWPL* model establishes an inventory of the software practices used by SMEs under assessment and initiates an approach for improving these practices. *OWPL* is based on both CMM and SPICE however it is a simplified model that takes SME specifics into account (e.g. small size, simple structure, limited number of versatile actors and a modest maturity level of processes).

Companies whose software maturity level is low use *OWPL* in order to highlight their strengths and weaknesses and to start a progressive and guided improvement approach.

Companies that already have a fair maturity level compared to their size and environment use *OWPL* in order to optimize their approach and, possibly get a maturity label from a CMM or SPICE accrediting body at lower cost.

*OWPL* not only contributes to improving software production processes in SMEs but also, thanks to the awareness effect it generates, *OWPL* helps improve the image of software production in the entire Industry making it more innovative and efficient.

This document is composed of four parts. Following this introduction, Section 2 describes the underlying logic leading to the elaboration of the model. Section 4 gives a detailed presentation of the model and Section 5 describes the methodology to follow during a software assessment and improvement exercise in industry.

## 2. Conceiving the OWPL model

The size and complexity of existing models like CMM and SPICE make their implementation excessively heavy and expensive for SMEs. The amount of information they contain and the number of processes and attributes they define make them unusable by small structures. Moreover, resorting to a company specialized in software process improvement is very expensive. According to a 1997 study<sup>[xviii]</sup> the cost of such an intervention can represent up to 50% of the annual budget for nearly 42% of the Walloon SMEs. Besides, the financial investment, the implementation of an approach such as CMM or SPICE requires people and resources at disposal. But according to the same study, 64% of SMEs mainly plan their activities on a daily base and 22% on a weekly base. In two companies out of three, information technologies are not formalized in company organization charts.

The simplicity of SME structures, the small size of the development teams and their way of working justify the need for customizing existing models to the particularities of SMEs.

OWPL was designed in order to highlight quickly the software practices that need improvement and to help draw a simple plan of action for improving those practices. OWPL also sets up a mean to take simple measures to study the variance induced by improvement actions enabling a company to assess its progression.

### 2.1. Tailoring constraints

The approach followed to construct the OWPL model and the associated evaluation method was based on the conclusions of two studies respectively using CMM<sup>[xix]</sup> and SPICE<sup>[xxiii]</sup> in small size data-processing structures.

The conclusions were the following ones:

- Need to take into account the particular context of Walloon SME by stressing the "evolution" and "improvement" aspects, the attribution of a quality label is not a priority,
- Need to use an accessible and unambiguous vocabulary that leaves no error of interpretation,
- Need to explain to SMEs the objectives of a quality improvement approach and show them where the potential returns on investment are,
- Need to define a methodology to standardize how the evaluation model is to be applied. The methodology should include explanation on the early informative stage where customer are explained the evaluation methodology as well as information on how the results of the evaluation will be communicated,
- Need to avoid bureaucracy and to limit the number of documents to the bare essential,
- No need to address the processes usually associated with higher level of maturity.

A first experiment in the field performed in the context of *micro evaluations of software practices*<sup>[xxiv]</sup> also help construct our approach based on concrete experiences.

## 2.2. Structuring the OWPL model

Practice shows that many companies have difficulties specifying the objectives of activities related to software practice improvement. It is therefore important for our methodology to show how goals of an approach to software-practice improvement relate to companies' business goals.

OWPL rests on the assumption that a business activity is carried out to help achieve one or more business goals in a direct or indirect way. Figure 1 shows the relationship between the goals of activities, processes, departments and enterprise. Each activity has its own objective that contributes to achieving the objective of the process to which it belongs. In turn, process goals are in line with the objectives of a particular department (e.g.: software processes). Finally, department objectives fulfill enterprise business goals. This hierarchical organization helps structure goals and it becomes easy to identify the relationships between goals at different level.

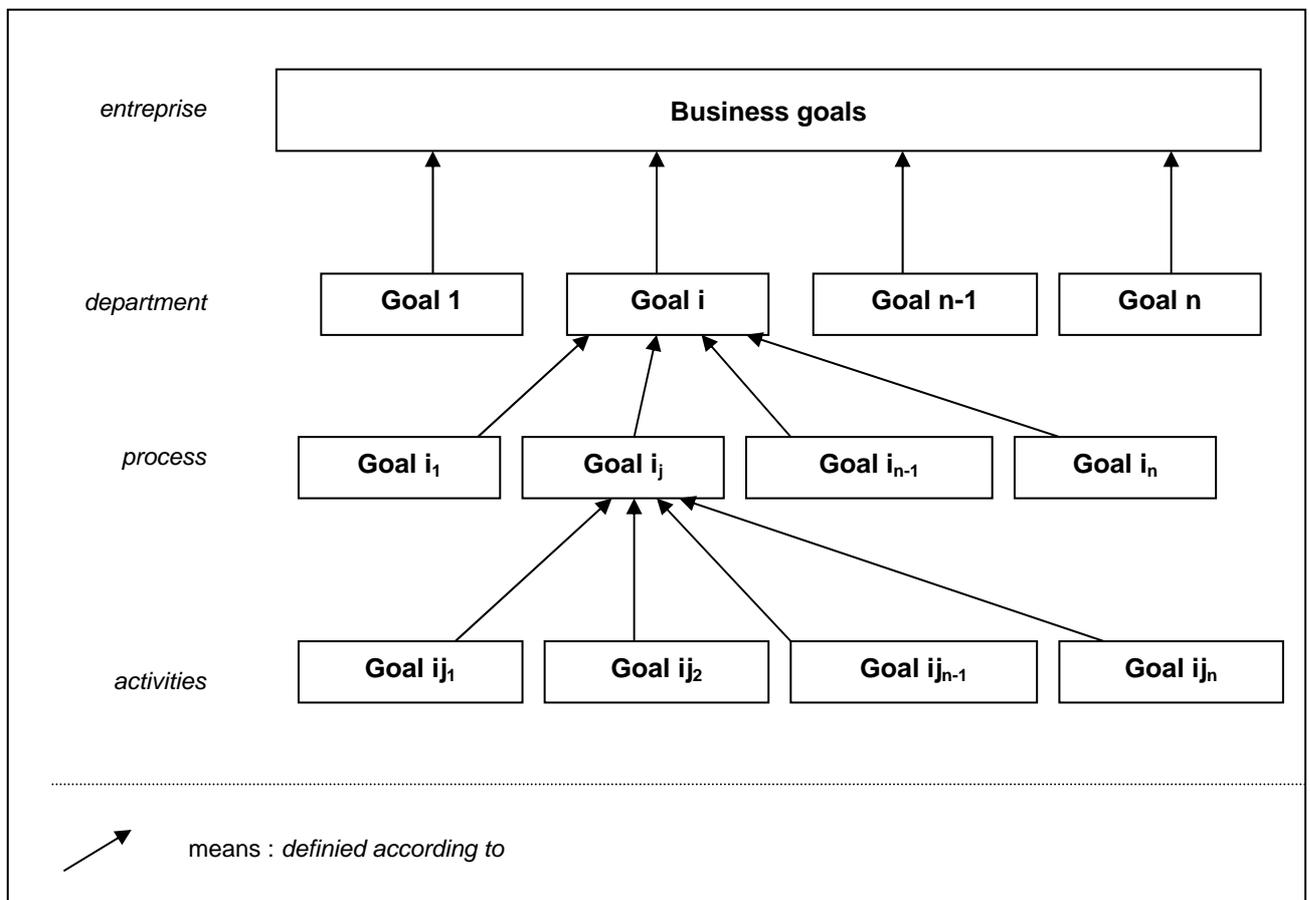


Figure 1: The objectives tree

Defining objectives is a critical step during an improvement approach. It helps concentrate efforts, motivate actors and control the efficiency of the improvement approach.

Moreover, objectives alone are not enough, a number of factors must be present in the enterprise in order to stabilize the environment and facilitate the implementation of processes and execution of activities. We call these factors *success factors*. The purpose of success factors is to show the level of resources availability, of management commitment, of an organization's effectiveness, and so on.

### 2.3. Structure of the OWPL model

The **Process** is the central component of the OWPL model.

As shown in Figure 1, a process is defined according to a specific process goal that transitively contributes to an enterprise's business goal. A process is specified to answer a company's concerns on a particular topic (e.g. development process or configuration management process). Each process is composed of **practices** contributing to its implementation. Practices are a set of related activities. In parallel, success factors support the proper implementation of a process and guarantee its efficiency.

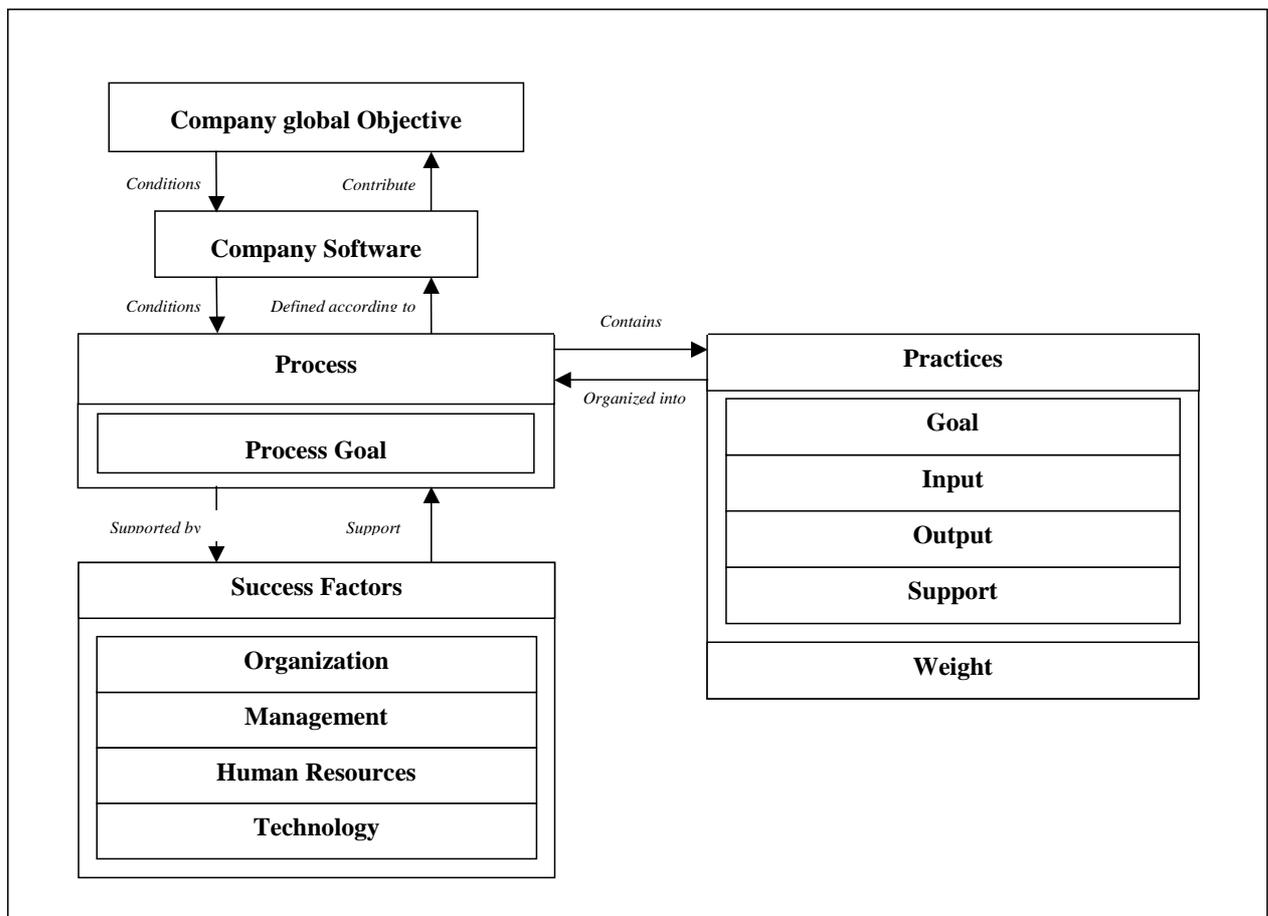


Figure 2 : OWPL model structure

### 2.3.1 Process

A process is a *structured set of necessary practices contributing to the achievement of a common and well-defined objective*.

A process is therefore defined by its objectives. In turn, we say that a process is composed of a set of practices that help achieve these objectives (see Figure 2: OWPL Model structure). It also transitively relates to a specific enterprise objective (see Figure 1: The objectives tree). Moreover, a process relies on *success factors* and it organizes its practices in a coherent way to achieve the specific enterprise objective.

### 2.3.2 Practices

A practice is *a set of software engineering or management activities that produces the output (work products) of a process or enhances the capability of a process* [xxvii].

Practices are organized in a way that facilitates their articulation around the process objective they belong to. Each practice is characterized by 5 attributes: its objective, inputs, outputs, resources and its weight in the achievement of the process objective. (see Figure 2: OWPL Model structure)

### 2.3.3 Success factors

Success factors are *environmental elements that support optimal process execution*.

There are four categories of success factors: the **organization** in which processes are executed, the **management** policy, the motivation of **human resources** and the **technical** means in place. (see Figure 2: OWPL Model structure)

### 3. Compatibility of OWPL with ISO/IEC12207

As explained in previous sections, OWPL model has been elaborated by extracting from existing standards CMM and ISO15504, the set of processes and practices that were appropriated for SME and VSE. Moreover, the terminology used in these standards has been simplified when it was relevant in order to avoid misunderstanding problem among SME and VSE.

Spontaneously, the reader or future user of OWPL model will ask if the proposed model maintained the compliance with source standards and furthermore is compliant with other widely used standards like ISO/IEC12207 or ISO/IEC90003 (Software and system engineering-Guidelines for the application of ISO 9001:2000 to computer software). Answer to these questions is positive since one of the objectives of the model is to serve as a first entry point in the difficult use and deployment of process standards.

However, OWPL model makes him stand out from ISO/IEC12207 and 90003 by its nature and its specific objectives. OWPL doesn't aims to set up and manage a quality management system but strongly focus on improve software practices among SME/VSE. A detailed mapping of OWPL model vs ISO/IEC12207 AMD1 & AMD2 can be found in annex.

SME/VSE are not obliged by OWPL model to formalize software processes and practices not suited to the company smooth running or objectives.

Therefore, using OWPL model can be considerate as an ISO/IEC9001:2000 certification preliminary step.

Moreover, an in-depth analysis of OWPL model shows that proposed processes largely cover requirements of the ISO/IEC9001:2000 standard. Process orientation of this new version of ISO/IEC9001:2000 reinforce the compatibility with the OWPL model.

In this context, OWPL model should be considered as an essential step for SME and VSE aiming to improve their software practices before an ISO/IEC9001:2000 certification.

## 4. Detailed presentation of OWPL

### 4.1. Process

As mentioned above, each process is defined by its objective and the list of its practices.

The order in which the processes and practices are presented below does not necessarily reflect their implementation order within the company. However, common sense shows that when information generated by a practice A are needed by another practice B then practice A must be implemented before B. In turn, if practices A and B belong to two different processes Proc<sub>A</sub> and Proc<sub>B</sub> respectively, then Proc<sub>A</sub> must be in place before Proc<sub>B</sub> can be executed.

OWPL is made up of the 10 following processes:

- Requirements management (**REQU**)
- Documentation (**DOCS**)
- Project planning (**PLAN**)
- Project tracking and oversight (**TROV**)
- Development (**DVLP**)
- Testing (**TEST**)
- Configuration management (**CONF**)
- Subcontract management (**SUMA**)
- Quality assurance (**QUAL**)
- Experience capitalization (**CPTL**)

In section 3.1, each process is described by its goal along with its description and a list of practices that compose the process. Then in section 3.2, we reiterate over each process, however this time, we focus on providing a systematic description of the practices composing the process. This systematic description of a practice provides the goal, inputs, outputs and resources needed by that practice.

## 4.1.1 Requirements Management - REQU

### **Goal**

The purpose of **Requirements Management** is to define customer requirements unambiguously, to ensure a common comprehension of requirements between the stakeholders, and to guarantee that requirements evolution is handled as part of the project.

### **Description**

Requirements Management includes the production and the maintenance of Software Requirements Specifications on the basis of customer demands and changes in these demands. Software Requirements Specifications will then constitute the basis for cost estimate, planning, implementation and tracking of activities throughout the project.

Requirements Management is one of the principal parameters for process stabilization and successful repeatability.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[REQU/PR01/03]	Analyzing requirements
[REQU/PR02/03]	Managing change requests
[REQU/PR03/03]	Validating the software product

## 4.1.2 Documentation - DOCU

### **Goal**

The purpose of **Documentation** is to develop and maintain the documents necessary to install, use and support the software product effectively.

### **Description**

Documentation requirements are listed in the Software Requirements Specifications documents. They specify the required documentation documents and describe the content of these documents. SRS documents may evolve during a project and affect the content of documentation. Thus, it is also part of the documentation process to check the correspondence between the content of documentation documents and their latest description in the SRS documents.

Documentation documents typically required are: general presentation documents, reference guides, support for training courses, installation and user guides. The documentation process is not concerned with the management of the documents produced by other processes. The management of such documents falls under the Configuration Management process.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[DOCU/PR01/03]	Identifying documentation requirements
[DOCU/PR02/03]	Developing documentation documents
[DOCU/PR03/03]	Providing the documentation documents to the appropriate people

### 4.1.3 Project Planning - PLAN

#### **Goal**

The purpose of **Project Planning** is to establish reasonable estimates of resources needed for tasks of development and of project management and also, to allocate the needed resources to these tasks.

#### **Description**

Project Planning provides necessary information to the project manager to organize the work of his/her team on the basis of objective data: resources necessary vs. resources available (time, financial, staff, technical), tasks to be carried out in-house and those to sub-contract, and the updated status of the project in progress. Planning gives an overview of the project and highlights the risk by showing the constraints associated with each task.

Planning is the basic mechanism of control and project management. The estimates derived from planning are crucial to good project management. It enables an effective monitoring of each project task and it highlights the eventual difficulties to respect engagements.

#### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[PLAN/PR01/06]	Decomposing the project in tasks
[PLAN/PR02/06]	Identifying constraints associated with a task
[PLAN/PR03/06]	Estimating the necessary resources
[PLAN/PR04/06]	Analyzing risks
[PLAN/PR05/06]	Creating the planning
[PLAN/PR06/06]	Adapting the planning

#### 4.1.4 Project Tracking and Oversight - TROV

##### **Goal**

The purpose of **Project Tracking and Oversight** is to check that the project proceeds in accordance with the planning estimates. This process tracks objectively and precisely project progress in order to detect possible variations from the estimates and, when necessary, takes the necessary corrective actions.

##### **Description**

This process consists in the permanent evaluation of the real progress of a project, of the resources used by a project, of the variations between what was planned and what is actually carried out, and finally, of the quantity of remaining work to carry out. Tracking activities must be carried out objectively and precisely for their results to be meaningful. Moreover, these results must help determine the origins of noted variations and also help to take adequate measures in order to reduce and prevent future variations.

The practices of **Project Tracking and Oversight** help gain visibility on the different project activities. The difficulties with respect to the engagements are identified as they emerge and the risks of major crisis are attenuated.

##### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[TROV/PR01/04]	Recording project progress
[TROV/PR02/04]	Analyzing project progress
[TROV/PR03/04]	Taking corrective decision
[TROV/PR04/04]	Conducting project assessment and wrap-up meeting

## 4.1.5 Development - DVLP

### **Goal**

The purpose of **Development** is to describe each of the stages related to the production of a software product, i.e., functional analysis, systems design and detailed design, coding and code review, debugging, install procedures and maintenance.

### **Description**

The development process integrates in a coherent way the practices to carry out to produce a software product. This process begins with the description of the software functionality based on customer requirements. From this description, the architecture of the system and a technical description of each component are derived. They are then translated into operational code. After elimination of errors and validation by the customer, the software produced is deployed. Thereafter starts the maintenance stage with corrections to the software product.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[DVLP/PR01/05]	Functional Analysis
[DVLP/PR02/05]	Design
[DVLP/PR03/05]	Coding
[DVLP/PR04/05]	Debugging
[DVLP/PR05/05]	Deployment and corrective maintenance

## 4.1.6 Testing - TEST

### **Goal**

The purpose of **Testing** is to check the adequacy of the software product compared to the requirements, and to detect as many errors as possible.

### **Description**

Testing plans are defined at the beginning of a project and are adapted (further specified) during each phase of the project development. They check if the software product respects customer requirements especially with respect to critical aspects of the product and if the software product is in accordance with predefined quality criteria.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[TEST/PR01/04]	Defining a test strategy
[TEST/PR02/04]	Defining test plan and creating test suite
[TEST/PR03/04]	Tracking and overseeing tests
[TEST/PR04/04]	Conducting tests

## 4.1.7 Configuration Management - CONF

### **Goal**

The purpose of **Configuration Management** is to establish and then maintain a coherent relationship between the different components of a project throughout the project's lifecycle. It consists in identifying the configuration of the software product at specific times, in controlling systematically the changes brought to the software configuration, and in maintaining the integrity and the traceability between the different elements of the software product.

### **Description**

Configuration Management determines

- All the elements of configuration of a software product,
- Their various versions,
- The relations between various applications at the level of libraries sharing and access to databases,
- The status of the software product, i.e., whether the product is in development, debug, test, or release phase.

Configuration management also distinguishes between products delivered to the customer including elements necessary to their creation such as source code and all other documents produced within the framework of project management. The former type of document may be seen by the customer whereas the latter stay private to the development firm.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[CONF/PR01/03]	Identifying work product
[CONF/PR02/03]	Identifying relationships between work products
[CONF/PR03/03]	Tracking the configuration

## 4.1.8 Subcontract Management - SUMA

### **Goal**

The purpose of **Subcontract Management** is to define contractually the reciprocal commitments the client and the subcontractor agree to, and to set up the structure to ensure the respect of the contract.

### **Description**

When managing subcontractors, an organization can then check the activities placed under the responsibility of a third party as well as monitor and possibly control subcontractors' performance with respects to contractual commitments.

Thanks to such a supervision and control, an organization can enforce the application of its processes to a subcontractor. In turn, this helps guarantee the quality of the end product once the work of the subcontractor is integrated.

Successful subcontract management makes the interventions of the subcontractor transparent to the final customer. Such transparency mitigates the risks related to customer/supplier relationships and promotes a successful ending to projects.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[SUMA/PR01/04]	Writing the invitation-to-tender
[SUMA/PR02/04]	Selecting subcontractor
[SUMA/PR03/04]	Tracking subcontract agreement (and requirement evolution)
[SUMA/PR04/04]	Receiving subcontracted work

## 4.1.9 Quality Assurance - QUAL

### **Goal**

The purpose of **Quality Assurance** is to ensure that the quality standards associated with each process are respected, and that a software product fulfills the quality requirements of the company.

### **Description**

Quality Assurance provides an adequate visibility on the processes executed during a project and on the work products created during a project lifecycle. The result of the project and the repeatability of the processes are better ensured when enforcing the procedures related to each process. In our context, a procedure is a verbal description of how a process is to be executed. The repeatability of processes ensures products of homogeneous quality, which contributes to reinforce customer satisfaction and confidence.

### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[QUAL/PR01/05]	Defining and adapting standards and procedures
[QUAL/PR02/05]	Monitoring product quality
[QUAL/PR03/05]	Monitoring Process quality
[QUAL/PR04/05]	Taking corrective decision
[QUAL/PR05/05]	Diffusing information

#### 4.1.10 Experience capitalization - CPTL

##### **Goal**

The purpose of **Experience Capitalization** is to generalize to all projects (in progress or future) the practices that supported the success of former projects.

##### **Description**

Practices that supported success are identified during the post mortem analysis of past projects. They are collected and saved in order to create a training framework. This in turn enables taking advantage of experiences successfully applied in former projects to new projects.

In the same way, practices which could be source of project failure are identified and drawn aside or adapted before reuse.

##### **Essential Practices**

<u>Reference</u>	<u>Name of the practice</u>
[CPTL/PR01/03]	Analyzing former projects
[CPTL/PR02/03]	Tailoring training framework
[CPTL/PR03/03]	Spreading continuously information on quality manual

## 4.2. Practices

A practice is a software engineering or management activity that contributes to the creation of process outputs (work products) or that enhances the capability of a process [xxvii].

A process is composed of several practices. On the other hand, *OWPL* was conceived in such a way that a practice only belongs to a single process. However, the execution of practice of a process may influence or may be a preliminary condition to the proper execution of another practice in the same or in other process.

Each practice is characterized by 5 attributes: its objective, inputs, outputs, resources and its weight in the achievement of the process objective. In this document, we only provide the first four attributes. From

### Practice goal

The goal of a practice is given as a short description including consequences, limitations and intention of the practice. An effective practice execution generates outputs needed to achieve the objective of the process to which the practice belongs.

### Practice inputs

The inputs of a practice are useful or essential elements needed to execute the practice. Those elements are usually produced by other practices. The proposed input list given below for each practice is not exhaustive and may be extended.

### Practice outputs

The outputs of a practice are concrete elements generated when executing the practice. Their existence confirms the accurate practice execution. The proposed output list given below for each practice is not exhaustive and may be extended.

### Practice resources

Resources represent technical elements, documents, tools or methods helping in executing the practice. The proposed resources list available for each practice is not exhaustive.

### Practice weight

An intrinsic (subjective) value allocated to a practice reflecting its importance as compare to the other practices of the same process. Together the weights of all practices making up a particular process add up to 100%.

**Remark:** The weights of practices are an instrument used during an evaluation hence they do not belong to the *OWPL* model per se. Thus, they are not presented in this document.

### Practice distribution by process

The following table shows how practices spread among the *OWPL* processes:

Process	Number of practices
Requirements Management - REQU	3
Documentation - DOCU	3
Project Planning - PLAN	6
Project Tracking and Oversight - TROV	4
Development - DVLP	5
Testing - TEST	4
Configuration Management - CONF	3
Subcontract Management - SUMA	4
Quality Assurance - QUAL	5
Experience Capitalization - CPTL	3
<b>Number of process : 10</b>	<b>40</b>

## 4.2.1 Requirements management - REQU

<b>Analyzing requirements</b>		<b>PROCESS : REQU</b> <b>PRACTICE : PR01/03</b> <b>REF : REQU/PR01/03</b>
<b>Goal</b>	<b>Elicit, clarify and define unambiguously customer requirements to guarantee a common understanding by all stakeholders in the project.</b>	
<b>Inputs</b>	Requirements	
<b>Outputs</b>	Requirements document Pre-study	
<b>Resources</b>	Negotiation techniques and techniques for conducting meetings Modeling tools	

<b>Managing change requests</b>		<b>PROCESS : REQU</b> <b>PRATIQUE : PR02/03</b> <b>REF : REQU/PR02/03</b>
<b>Goal</b>	<b>Maintain the consistency between requirements documents and evolving requirements throughout the whole project.</b>	
	Customer can introduce new requirements during a project or initial requirements may become obsolete. The supplier, given his technical knowledge, can also influence customer's choices for development environment, tool, method, graphical user interface of the software product being developed, and so on.	
<b>Inputs</b>	Requirements document Pre-study New requirement or requirement change request	
<b>Outputs</b>	Requirements document Amendment to the requirements document	
<b>Resources</b>	Negotiation techniques, techniques for conducting meeting Modeling tools	

<b>Validating the software product</b>		<b>PROCESS : REQU</b> <b>PRATIQUE : PR03/03</b> <b>REF : REQU/PR03/03</b>
<b>Goal</b>	Assess the software product to determine whether it fulfills customer requirements. This practice consists in conducting system and acceptance testing.	
<b>Inputs</b>	Software product Requirements document Pre-study Acceptance test report	
<b>Outputs</b>	Assessment report	
<b>Resources</b>	Execution environment	

## 4.2.2 Documentation - DOCS

<b>Identifying documentation requirements</b>	<b>PROCESS : DOCS</b> <b>PRATICE : PR01/03</b> <b>REF : DOCS/PR01/03</b>
<p><b>Objective</b> Elicit, clarify and define unambiguously the documentation documents to produce and specify precisely their content.</p> <p><b>Inputs</b> Requirements document Functional Analysis Planning</p> <p><b>Outputs</b> Listing of the documents to produce with a description of their content Documents distribution scenarios</p> <p><b>Resources</b> Modeling tools</p>	

<b>Developing documentation documents</b>	<b>PROCESS : DOCS</b> <b>PRATICE : PR02/03</b> <b>REF : DOCS/PR02/03</b>
<p><b>Objective</b> Write documentation documents that guaranty an optimal installation, use, and support of the software product.</p> <p><b>Inputs</b> Listing of the documents to produce with a description of their content Documentation templates Documentation norms and standards The software product Functional Specifications Requirements document Tests reports System and software architectures Technical descriptions of the components Integration plan</p> <p><b>Outputs</b> Overall introduction document Online help Reference manual Installation manual Usage manual Training support</p> <p><b>Resources</b> Development environment Office software tools</p>	

**Providing the documentation documents to the appropriate people**

**PROCESS : DOCS  
PRATICE : PR03/03  
REF : DOCS/PR03/03**

<b>Objective</b>	Provide to appropriate people or organization the needed information on how to install, use and support the software product.
<b>Inputs</b>	Listing of the documentation documents with a description of their content Documents distribution scenarios Overall introduction document Online help Reference manual Installation manual Usage manual Training support
<b>Outputs</b>	Report on the distribution mechanisms for the documentation documents
<b>Resources</b>	Configuration management Email tools Intranet

### 4.2.3 Projct Planning- PLAN

<b>Decomposing the project in tasks</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR01/06</b> <b>REF : PLAN/PR01/06</b>
<b>Objective</b>	Partition the project in work units in order to estimate as precisely as possible resource consumption.	
<b>Inputs</b>	Requirements document Pre-study	
<b>Outputs</b>	Listing of tasks to accomplish	
<b>Resources</b>	Project management tool	

<b>Identifying constraints associated to a task</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR02/06</b> <b>REF : PLAN/PR02/06</b>
<b>Objective</b>	Organize tasks based on their associated constraints.	
<b>Inputs</b>	Listing of tasks	
<b>Outputs</b>	Listing of constraints Organized, prioritized listing of tasks	
<b>Resources</b>	Project management tool	

<b>Estimating the necessary resources</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR03/06</b> <b>REF : PLAN/PR03/06</b>
<b>Objective</b>	Identify the quantities of resources needed to conduct the project	
<b>Inputs</b>	Listing of tasks Listing of constraints	
<b>Outputs</b>	Resources needed per tasks	
<b>Resources</b>	Estimation tools Project management tool	

<b>Analyzing risks</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR04/06</b> <b>REF : PLAN/PR04/06</b>
<b>Objective</b>	Identify tasks whose execution may no go according to plan and determine the measures to take when such a situation occurs to avoid letting the rest of the project slip out of schedule.	
<b>Inputs</b>	Listing of tasks Listing of constraints Needed resources per tasks Overall availability of resources	
<b>Outputs</b>	Risk analysis documents	
<b>Resources</b>	Project management tool Risk management tool	

<b>Creating the planning</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR05/06</b> <b>REF : PLAN/PR05/06</b>
<b>Objective</b>	Arrange all tasks of a project to optimize the resource usage and limit the risk factors.	
<b>Inputs</b>	Listing of tasks Listing of constraints Needed resources per task Overall availability of resources Risk analysis documents	
<b>Outputs</b>	Planning Overall resources needed Listing tasks to subcontract	
<b>Resources</b>	Project management tool Risk management tool	

<b>Adapting the planning</b>		<b>PROCESS : PLAN</b> <b>PRATICE : PR06/06</b> <b>REF : PLAN/PR06/06</b>
<b>Objective</b>	Adapt the planning according to real-life circumstances in which the project takes place.	
<b>Inputs</b>	Listing of tasks Listing of constraints Needed resources per tasks Overall availability of resources Risk analysis documents Corrective actions taken by project management	
<b>Outputs</b>	Risk analysis documents Updated planning Overall resources needed Listing tasks to subcontract	
<b>Resources</b>	Project management tool Risk management tool Estimation tools Project management and follow-up	

#### 4.2.4 Project Tracking and Oversight - TROV

<b>Recording project progress</b>		<b>PROCESS : TROV</b> <b>PRATICE : PR01/04</b> <b>REF : TROV/PR01/04</b>
<b>Objective</b>	Gather data in order to estimate objectively and precisely project progression, i.e., <i>what is done vs. what is left.</i>	
<b>Inputs</b>	Planning Deliverables accomplished or being accomplished	
<b>Outputs</b>	Status of project progression	
<b>Resources</b>	Measuring tools Method for configuration management Configuration management tool Formal reviews	

<b>Analyzing project progress</b>		<b>PROCESS : TROV</b> <b>PRATICE : PR02/04</b> <b>REF : TROV/PR02/04</b>
<b>Objective</b>	Compare the current status ( <i>what is done vs. what is left</i> ) of the project with the previously established planning to detect potential deviation.	
<b>Inputs</b>	Status of project progression	
<b>Outputs</b>	Noted deviation from the planning	
<b>Resources</b>	Method of project management Resource estimation	

<b>Taking corrective decisions</b>		<b>PROCESS : TROV</b> <b>PRATICE : PR03/04</b> <b>REF : TROV/PR03/04</b>
<b>Objective</b>	Enumerate the corrective actions to take in order to avoid further delays and when possible, to resorb the current deviation	
<b>Inputs</b>	Planning Listing of tasks Listing of constraints Risk analysis documents Noted deviation from the planning	
<b>Outputs</b>	Corrective actions	
<b>Resources</b>	Method of project management Resource estimation	

**Conducting project assessment and wrap-up meeting****PROCESS : TROV  
PRATICE : PR04/04  
REF : TROV/PR04/04**

<b>Objective</b>	At the end of the project, conduct an a posteriori analysis on how the project ran in order to capitalize on experiences (positive and negative) for future projects.
<b>Inputs</b>	Planning Listing of tasks Listing of constraints Risk analysis documents Noted deviation from the planning Historic of noted deviations from the planning Historic of the corrective actions taken
<b>Outputs</b>	Project assessment
<b>Resources</b>	Coordination meeting Technical audit

## 4.2.5 Development - DVLP

<b>Functional Analysis</b>		<b>PROCESS : DVLP</b> <b>PRATICE : PR01/05</b> <b>REF : DVLP/PR01/05</b>
<b>Objective</b>	Produce a precise description of the software functionality for the software product to develop	
<b>Inputs</b>	Requirements document Pre-study (Tests report)	
<b>Outputs</b>	Functional specifications Functional test suite	
<b>Resources</b>	Methods of analysis	

<b>Conception</b>		<b>PROCESS : DVLP</b> <b>PRACTICE : PR02/05</b> <b>REF : DVLP/PR02/05</b>
<b>Objective</b>	Design system components and give a technical description for each of them.	
<b>Inputs</b>	Requirements document (Tests report)	
<b>Outputs</b>	System and software architectures Technical description of components Integration Plan	
<b>Resources</b>	Specification methodology Design and specification modeling tools	

<b>Coding</b>		<b>PROCESS : DVLP</b> <b>PRACTICE : PR03/05</b> <b>REF : DVLP/PR03/05</b>
<b>Objective</b>	Translate the technical description of components in (source) code for the selected operating system(s).	
<b>Inputs</b>	System and software architecture Technical description of components (Test reports)	
<b>Outputs</b>	Documented code Unit test suite	
<b>Resources</b>	Development environment Programming method and standards	

<b>Debugging</b>	<b>PROCESS : DVLP PRACTICE : PR04/05 REF : DVLP/PR04/05</b>
<p><b>Objective</b>      Detect and correct bugs in the source code while developing the software product.</p> <p><b>Inputs</b>          Documented code</p> <p><b>Outputs</b>        Operational and documented code</p> <p><b>Resources</b>      Debugging method and tools Development Environment</p>	

<b>Deployment and corrective maintenance</b>	<b>PROCESS : DVLP PRACTICE : PR05/05 REF : DVLP/PR05/05</b>
<p><b>Objective</b>      Deploy the software product at the client site, solve installation problems and other errors not caught during prior testing.</p> <p><b>Inputs</b>          Software product</p> <p><b>Outputs</b>        Installation report Error report</p> <p><b>Resources</b>      Operating system(s) Development environment Test Environment and tools</p>	

## 4.2.6 Tests - TEST

<b>Defining a test strategy</b>	<b>PROCESS : TEST</b> <b>PRACTICE : PR01/04</b> <b>REF : TEST/PR01/04</b>
<p><b>Objective</b>      Develop test plans that describe how tests execution helps reach the objectives set for each development phase and how tests execution ensures that the final software product will fulfill customer requirements.</p> <p><b>Inputs</b>            Requirements document  Pre-study  Planning  Listing of tasks</p> <p><b>Outputs</b>            Integration test plan  System test plan</p> <p><b>Resources</b></p>	
<b>Creating test suites</b>	<b>PROCESS : TEST</b> <b>PRACTICE : PR02/04</b> <b>REF : TEST/PR02/04</b>
<p><b>Objective</b>      Create test suites for each development phase in order to verify when a development phase satisfies its requirements as stated at the beginning of the phase or as described in institutionalized test procedures.</p> <p><b>Inputs</b>            System test plan  Functional specifications  System Architecture  Technical Description of components</p> <p><b>Outputs</b>            Acceptance and System test suite  Unit test suite</p> <p><b>Resources</b></p>	
<b>Tracking and overseeing tests</b>	<b>PROCESS : TEST</b> <b>PRACTICE : PR03/04</b> <b>REF : TEST/PR03/04</b>
<p><b>Objective</b>      This practice has two objectives: first, to maintain consistent relationships between test plans, test suites and requirements and second, to verify that problems highlighted by earlier tests have been solved.</p> <p><b>Inputs</b>            Requirements documents  Pre-study  Amendment to the requirements document  Planning  Listing of tasks</p> <p><b>Outputs</b>            Integration test plan  System test plan</p> <p><b>Resources</b></p>	

<b>Conducting tests</b>	<b>PROCESS : TEST PRACTICE : PR04/04 REF : TEST/PR04/04</b>
<b>Objective</b>	Verify that the software product fulfills the requirements and detect defects, and when possible highlights the cause of defects.
<b>Inputs</b>	Operational code Integration test plan Acceptance test suite Unit test suite System test suite
<b>Outputs</b>	Test report Software product
<b>Resources</b>	Test method and tools Test Environment Office tool suite

## 4.2.7 Configuration management- CONF

<b>Identifying work products</b>	<b>PROCESS : CONF</b> <b>PRACTICE : PR01/03</b> <b>REF : CONF/PR01/03</b>
<p><b>Objective</b> Identify and save all versions of all work products (software, hardware, documentation, analysis documents, etc) resulting from each practice.</p> <p><b>Inputs</b> Planning Functional Specifications Deliverables</p> <p><b>Outputs</b> Listing of the configuration elements</p> <p><b>Resources</b> Configuration management tool</p>	
<b>Identifying relationships between work products</b>	<b>PROCESS : CONF</b> <b>PRACTICE : PR02/03</b> <b>REF : CONF/PR02/03</b>
<p><b>Objective</b> Identify and save the relationships between configuration elements, for example, maintain the traceability between a requirement change and the corresponding amendments to test plans.</p> <p><b>Inputs</b> Listing of the configuration elements System and software architecture Technical description of components</p> <p><b>Outputs</b> Listing of the relationships between the configuration elements</p> <p><b>Resources</b> Configuration management tool Requirements tools (e.g. to trace requirements)</p>	
<b>Tracking the configuration</b>	<b>PROCESS : CONF</b> <b>PRACTICE : PR03/03</b> <b>REF : CONF/PR03/03</b>
<p><b>Objective</b> Guaranty the integrity of each version of the software product by maintaining current the relationships between the configuration elements.</p> <p><b>Inputs</b> Listing of configuration elements Listing of relationships between configuration elements System and software architecture Technical Description of components New configuration elements New relationships between configuration elements</p> <p><b>Outputs</b> Listing of relationships between configuration elements</p> <p><b>Resources</b> Configuration management tool Requirements tools (e.g. to trace requirements)</p>	

## 4.2.8 Subcontract management - SUMA

<b>Writing the invitation-to-tender</b>	<b>PROCESS : SUMA</b> <b>PRACTICE : PR01/04</b> <b>REF : SUMA/PR01/04</b>
<p><b>Objective</b> Identify the work to subcontract and its limits. Define the relationships and interfaces between the subcontracted work and the rest of the project. Define the rules of engagement between the subcontractor and your organization.</p> <p><b>Inputs</b> Requirements document Pre-study Functional Specifications Listing of tasks to subcontract</p> <p><b>Outputs</b> Subcontract-requirements document Invitation-to-tender for subcontract</p> <p><b>Resources</b> Technical negotiation, meeting techniques Modeling tool Resource Estimation</p>	
<b>Selecting subcontractors</b>	<b>PROCESS : SUMA</b> <b>PRACTICE : PR02/04</b> <b>REF : SUMA/PR02/04</b>
<p><b>Objective</b> Selecting the subcontractor and finalize together a collaboration agreement.</p> <p><b>Inputs</b> Selection criteria Selection procedure Subcontract-Requirements document Subcontract offers</p> <p><b>Outputs</b> Contract with subcontractor</p> <p><b>Resources</b> Decision tool Negotiation Method Tool to evaluate and compare subcontractors' capacity</p>	
<b>Tracking subcontract agreement (and requirement evolution)</b>	<b>PROCESS : SUMA</b> <b>PRACTICE : PR03/04</b> <b>REF : SUMA/PR03/04</b>
<p><b>Objective</b> Verify that the subcontractor fulfills its engagements, manages its work and follows up on the potential evolution in requirements.</p> <p><b>Inputs</b> Subcontract-requirements document Contract with subcontractor Planning New requirements</p> <p><b>Outputs</b> Follow-up report Amendment to the subcontract-requirements document</p> <p><b>Resources</b> Coordination meeting</p>	

<b>Receiving subcontracted work</b>	<b>PROCESS : SUMA PRACTICE : PR04/04 REF : SUMA/PR04/04</b>
<b>Objective</b>	Validate the subcontractor's work as compared to the subcontract-requirements document and contract. When validated, free the subcontractor from any further obligations.
<b>Inputs</b>	Subcontract-requirements document Contract with subcontractor Follow-up report Subcontractor's work and product
<b>Outputs</b>	Validation report of the subcontractor's work Validation and acceptance of the subcontractor's product
<b>Resources</b>	Test environment Test methods

## 4.2.9 Quality Assurance - QUAL

<b>Defining and adapting standards, norms and procedures</b>	<b>PROCESS : QUAL</b> <b>PRACTICE : PR01/05</b> <b>REF : QUAL/PR01/05</b>
<p><b>Objective</b>      Produce a set of references accessible to all parties involved in the execution of processes.</p> <p><b>Inputs</b>          End of project assessment Memorandums</p> <p><b>Outputs</b>        Standard norms et procedures Standard Documents / Template Programming standards Processes Description</p> <p><b>Resources</b>      Office Software tools Standard norms et procedures</p>	
<b>Monitoring product quality</b>	<b>PROCESS : QUAL</b> <b>PRACTICE : PR02/05</b> <b>REF : QUAL/PR02/05</b>
<p><b>Objective</b>        Verify that the software product fulfills the quality requirements of the company developing it.</p> <p><b>Inputs</b>            Software product Standard norms Tests report</p> <p><b>Outputs</b>          Quality assessment report</p> <p><b>Resources</b>        Inspection, review, audit Standard norms and procedures</p>	
<b>Monitoring processes quality</b>	<b>PROCESS : QUAL</b> <b>PRACTICE : PR03/05</b> <b>REF : QUAL/PR03/05</b>
<p><b>Objective</b>        Verify that standards, norms and procedures currently in place in the company were followed when developing the software product.</p> <p><b>Inputs</b>            Standard norms and procedures Quality manual Standard Documents / Templates Programming standards Processes Description Configuration elements</p> <p><b>Outputs</b>          processes quality assessment report</p> <p><b>Resources</b>        Inspection of configuration elements Review, audit</p>	

<b>Taking corrective decisions</b>	<b>PROCESS : QUAL PRACTICE : PR04/05 REF : QUAL/PR04/05</b>
<p><b>Objective</b> Define or amend standards, norms and procedures to include potential corrective actions to take when a project deviates from estimates established in the project planning.</p> <p><b>Inputs</b> Processes assessment report</p> <p><b>Outputs</b> Corrective actions implemented</p> <p><b>Resources</b> Coordination meetings</p>	

<b>Evaluating work and team members</b>	<b>PROCESS : QUAL PRACTICE : PR05/05 REF : QUAL/PR05/05</b>
<p><b>Objective</b> Motivate team members by providing them with a relevant, objective feedback on the quality of work they achieved on a just-completed project.</p> <p><b>Inputs</b> Software product End of project assessment Tests report Quality assessment report</p> <p><b>Outputs</b> Information Report</p> <p><b>Resources</b> Information meetings, emails, note posting, memorandums</p>	

#### 4.2.10 Experience Capitalization - CPTL

<b>Analyzing former projects</b>		<b>PROCESS : CPTL</b> <b>PRACTICE : PR01/03</b> <b>REF : CPTL/PR01/03</b>
<b>Objective</b>	Capitalize on experience acquired during previous projects.	
<b>Inputs</b>	Processes description End-of-project assessment Standard Procedures	
<b>Outputs</b>	Practices, procedures, rules to generalize	
<b>Resources</b>	Meetings Control	

<b>Tailoring the training framework</b>		<b>PROCESS : CPTL</b> <b>PRACTICE : PR02/03</b> <b>REF : CPTL/PR02/03</b>
<b>Objective</b>	Produce and manage a set of references accessible to all parties involved in projects.	
<b>Inputs</b>	Standard norms et procedures Processes description Practices to generalize	
<b>Outputs</b>	Quality manual Processes description	
<b>Resources</b>	Coordination meetings Office software suite, intranet	

<b>Spreading continuously information on quality manual</b>		<b>PROCESS : CPTL</b> <b>PRACTICE : PR03/03</b> <b>REF : CPTL/PR03/03</b>
<b>Objective</b>	Verify that the means of communications and the communication channels are in place for a timely communication of all modifications performed on quality assurance manuals to all parties involved.	
<b>Inputs</b>	Quality manual	
<b>Outputs</b>	Information Diffusion Report	
<b>Resources</b>	Various means of communication: email, memorandum, note posting Training	

### **4.3. Success factors**

Processes are defined according to an organization's business goal (see figure 2). Implementation of a process is performed mainly by applying the practices associated to that process; however, that implementation is heavily influence by environmental elements present in an organization. We call these environmental elements: *success factors*.

Success factors are defined as *environmental elements (organization, management, technology, human resources) supporting processes execution. Success factors don't produce project outputs or artefacts instead, they influence processes execution, in particular, they improve the impact and efficiency processes have on product quality.*

So, an efficient organization, a good management, an adequate technology and good employees enable better process execution, which in turn yields products of higher quality.

#### **Organization**

The processes must efficiently fulfil their mission for an organization to achieve its defined business goals. To do so, the organizational structure must be laid in a way that facilitates process execution.

The organizational structure must facilitate information flow and communication with the internal and external environment. Furthermore, the organizational structure must provide the necessary resources, and as important, an appropriate work environment. A good organizational structure helps define adequate goals and responsibilities, and promotes decision-making leading to business goal achievement.

#### **Management**

A good organizational structure enables efficient management of available resources and ensures an effective process execution.

To achieve its goal, the management of an organization must determine how to manage a project. It is accomplished by defining procedures to follow, processes to execute and standards to comply to. The set of procedure and standard must include resources allocation, the definition and implementation of controls mechanisms and also the decision-making mechanisms to correct potential drifts during a project.

#### **Human resources**

Humans are the most complex resources to manage within an organization. The successful implementation of processes heavily depends on the availability of the necessary human resources, their adequacy, and their potential and effective productivity.

*Tracking and oversight* practices are distributed throughout several processes. To guarantee the traceability of these practices, responsibilities must be assigned so the integrity and objectivity of tracking and oversight activities can be checked.

#### **Technology**

Processes can not be executed without proper tools support. Performance and adequacy of tools in place directly impact processes execution and goals achievement.

Sections 3.3.1, 3.3.2, 3.3.3 and 3.3.4 respectively describe the characteristics success factors highlights about the organizational, the managerial, the human and the technical environments.

### 4.3.1 Organization

<b>Organizational structure clearly defined</b>	REF : FS-ORG01/13
Existing organizational structure limits bureaucracy, facilitates information flow and helps decision-making.	
<b>Business goals clearly defined</b>	REF : FS-ORG02/13
The business goals of an organization are defined clearly and precisely so as to ease the specification of processes goals and increase coherence in decision-making.	
<b>Business goals known and accepted by all</b>	REF : FS-ORG03/13
Business goals are presented and explained to everyone in the organization to ensure collaboration and minimize potential resistance to change.	
<b>Management commitment to business goals</b>	REF : FS-ORG04/13
Management shows its commitment to business goals by supporting continuously the different departments in their actions related to reaching these business goals.	
<b>Processes goals clearly defined</b>	REF : FS-ORG05/13
Processes goals are derived from business goals, they are defined clearly and precisely to ensure comprehension by all people involved in executing these processes.	
<b>Processes goals known and accepted by all</b>	REF : FS-ORG06/13
Processes goals are presented and explained to all the people concerned in order to ensure full collaboration and to minimize potential resistance to change.	
<b>Management commitment to processes goals</b>	REF : FS-ORG07/13
Management publicly supports the goals of the various processes implemented in the organization and shows its commitment to the processes by enforcing their respect.	
<b>Communication with the internal environment</b>	REF : FS-ORG08/13
Communication channels in the organization are defined adequately for information to flow horizontally between services and departments, and vertically, between the various levels in the organizational structure of the company.	
<b>Communication with the external environment</b>	REF : FS-ORG09/13
Communication channels in the organization are defined adequately for information to flow with (from/to) the external environment (customers, suppliers, public authorities, etc...).	

<b>Necessary resources provided</b>	<b>REF : FS-ORG10/13</b>
Necessary resources are provided taking into account the project objectives to reach and the project constraints to respect.	
<b>Necessary work environment provided</b>	<b>REF : FS-ORG11/13</b>
Work environment and equipment at the disposal of everyone are adapted to the tasks to perform and take into account the objectives to reach and the constraints to respect.	
<b>Responsibilities clearly defined</b>	<b>REF : FS-ORG12/13</b>
Everyone's roles and responsibilities are officially defined and are re-examined regularly according to the evolution of projects and context.	
<b>Responsibilities well-known</b>	<b>REF : FS-ORG13/13</b>
Everyone's roles and responsibilities are communicated to all the people concerned (directly or not). Moreover the definition of responsibilities and their assignment can easily be looked up.	

### 4.3.2 Management

<b>Procedure definition</b>	REF : FS-MAN01/08
Procedures are defined in a clear, precise and unambiguous way understandable by all people involved in executing them	
<b>Procedures phases definition</b>	REF : FS-MAN 02/08
Each phase (or step) of a procedure is clearly and precisely defined so the execution of the procedure can be controlled effectively.	
<b>Definition of standards</b>	REF : FS-MAN 03/08
The qualitative and quantitative standards to comply with are defined in a clear and precise way. Moreover, the standards are communicated to the people concerned.	
<b>Review for standards compliance</b>	REF : FS-MAN 04/08
Review of compliance to standards and tasks to perform during such a review are defined in a clear and precise way so review activities can be controlled and verified.	
<b>Continuous tracking of procedure execution</b>	REF : FS-MAN05/08
Homogeneous execution of a procedure is systematically controlled by verifying the uniform application of practices related to the procedure.	
<b>Continuous tracking of standard compliance</b>	REF : FS-MAN06/08
Compliance to standards is systematically controlled by verifying uniform application of standards across projects and departments.	
<b>Corrective decision-making process</b>	REF : FS-MAN07/08
Management promotes corrective decision making by encouraging first, the identification of variations in planning as well as causes for these variations and second, the recordings of the findings to prevent reoccurrences in the same or other projects.	
<b>Resources allocation</b>	REF : FS-MAN08/08
Management allocates the necessary resources to projects and departments in an unbiased way and openly communicates its resource allocation decisions.	

### 4.3.3 Human resources

<b>Regular inventory of internal available skills</b>	REF : FS-HUM01/06
An inventory of in-house competences available is held up-to-date. This inventory is used to optimize resources allocation while taking into account the opinion and expectations of the people concerned as well as the constraints of the various projects.	
<b>Regular inventory of market available skills</b>	REF : FS-HUM 02/06
An inventory of competences available on the market is held up-to-date in order to acquire competences missing in-house through formation, recruitment, and consultancy	
<b>Regular assessment of in-house skills adequacy</b>	REF : FS-HUM 03/06
In-house skill adequacy is assessed regularly in order to decrease the risk of frustration in the event of underutilization or to avoid situation where an employee feeling incompetent because of a mismatch between the employee's skills and a task assigned.	
<b>Organization of training courses</b>	REF : FS HUM 04/06
Training courses are organized in order to ensure the adequacy of competences available in-house with regards to the tasks to realize, the procedures to respect, and the standards to comply with.	
<b>Employees motivation support</b>	REF : FS-HUM 05/06
The organization encourages employees to communicate their expectations and conversely, employees feel they are listened to. Management is concerned with employees' welfare and it involves employees at the various levels of the organization to maintain motivation high.	
<b>Independence and objectivity</b>	REF : FS-HUM 06/06
The practices (in particular practices of tracking and oversight) are assigned to people whose position and function guarantee independence and objectivity of results.	

### 4.3.4 Technology

<b>Regular inventory of technologies used in-house</b>	REF : FS-TEC01/03
An inventory of technologies used in-house is held up-to-date so as to optimize the use of technologies as well as to compare them with other technologies available on the market.	
<b>Regular inventory of technologies available on the market</b>	REF : FS-TEC02/03
An inventory of technologies available on the market is held up-to-date so as to evaluate their relevance compared to technologies used in-house.	
<b>Regular review of the relevance of technologies used in-house</b>	REF : FS-TEC03/03
Technologies used in-house are regularly reviewed in order to determine whether projects are well assisted by these technologies or whether new technologies should be identified and deployed.	

## 5. OWPL software practices assessment

OWPL was designed to quickly identify practices related to software development in need of improvement, and also, to help draw a simple plan of action aiming at improving those practices and measuring the improvements.

Software practices assessment according to OWPL takes into account the particularities of Walloon SME with their associated constraints (simplicity of enterprises structure, small size of software development teams, limited resources available). Performing an assessment must therefore limit the consumption of resources needed from the company assessed by taking little time for interviews and by avoiding the mobilizing of too many employees at once. Yet, the assessment must also make sure all pertinent information is collected.

OWPL assessment is not an end in itself but a mean of improving processes related to software development. An OWPL assessment identifies potential for evolution and improvement within the assessed unit. In turn, it is possible for an organization to be assessed several times and monitor its progress between several assessments.

### 5.1. OWPL assessment method

Usually, an OWPL assessment only focuses on some of the processes, which are selected on the basis of a former assessment such as a micro evaluation or of a quick interview prior to conducting the OWPL assessment. Focus on particular processes can also depend on explicit demand of the enterprise to assess or on the results of the enterprise value chain analysis.

The evaluation of the maturity of a process is a bottom-up computation. First the maturity level of each practice is assessed according to the quality of its content and its institutionalization within the enterprise. Then, the maturity of a process is estimated from aggregate maturity level of its practices.

Success factors supporting processes are only assessed according to their perceived quality, i.e., employee are asked their impression on particular environmental elements.

### 5.2. Evaluation of Practices

The practices are evaluated according to their description in the OWPL model here above. An evaluation checks if each practice produces the outputs it is supposed to, draws attention on how these outputs are produced (maturity level), and also identifies whether the practice is applied on all or only some projects (institutionalization degree).

### 5.3. Success factors evaluation

Success factors are evaluated for the organization as a whole and not on a per-employee basis. They are evaluated according to their description in the OWPL model. The evaluation of success factors determines whether processes receive the necessary support for their proper execution.

### 5.4. Results interpretation

Scores and interpretation method<sup>[x]</sup> have been empirically defined on the basis of results obtained from the first set of evaluations performed (by the same set of assessors).

### 5.5. Evaluation questionnaire

The questionnaire used for evaluations is derived from the OWPL model described in this document. However, the questionnaire is NOT part of the OWPL model <sup>[x]</sup>.

## 6. Bibliography

- [i] BRODMAN J. G. & JOHNSON D. L., *What Small Businesses and Small Organisations say about CMM ?* dans Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italie, Mai 1994.
- [ii] BROOKS F.P., *No Silver Bullet: Essence and Accidents of Software engineering*, IEEE Computer, Vol. 20, No. 4, Avril 1987, pp. 10-19.
- [iii] BROOKS F.P., *The Mythical Man-Month*, Essays on Software Engineering, Anniversary Edition, Addison-Wesley, 1995.
- [iv] CAPUTO K., *CMM : Implementation Guide Choreographing Software Process Improvement*, Addison-Wesley, 1998.
- [v] DEMING W. E., *Out of the crisis*, MIT, 1997.
- [vi] DYMOND K.D., *Le guide du CMM : Introduction au modèle de maturité CMM*, Cepadues, 1997.
- [vii] EI EMAN K., DROUIN J.-N. & Melo, W., *SPICE : The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE, COMPUTER SOCIETY, 1998.
- [viii] FAIRLEY R. E., *Software Engineering Concepts*, McGraw-Hill International, 1985.
- [ix] HABRA N., NIYITUGABIRA E. & RENAULT A., *Modèle OWPL - Glossaire*, OWPL-FUNDP 1999
- [x] HABRA N., NIYITUGABIRA E. & RENAULT A., *Modèle OWPL - Questionnaire*, OWPL-FUNDP 1999
- [xi] HABRA N. & RENAULT A., *Modèle OWPL - Méthode de cotation*, OWPL-FUNDP 2000
- [xii] HUMPHREY W. S., *A Discipline for Software Engineering*, Addison-Wesley, 1995.
- [xiii] HUMPHREY W. S., *Introduction to the Personal Software Process*, Addison-Wesley, 1997.
- [xiv] HUMPHREY W.S., *Managing the Software Process*, SEI Series in Software Engineering, Addison-Wesley, 1991.
- [xv] ISO/IEC JTC 1/SC 7, ISO/IEC TR 15504, 1998.
- [xvi] JOHNSON D.L. & BRODMAN J.G., *Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects*, Software Process Newsletter, N° 8, IEEE, Winter 1997.
- [xvii] KOCH G., *Process Assessment : the Bootstrap Approach*, Information and Software Technology, Vol30, N°6/7, 1993.
- [xviii] LOBET-MARIS Cl., *Utilisation des systèmes d'information interorganisationnels par les PME belges - Rapport final - CITA-FUNDP - novembre 1997.*
- [xix] NIYITUGABIRA E., *Adéquation du modèle CMM aux PME et pistes d'adaptation*, OWPL-FUNDP, juillet 1998.
- [xx] PAULK M. C., CURTIS B., CHRUSIS M. B. & WEBER C., *Capability Maturity Model for Software, Version 1.1*, SEI, CMU/SEI-93-TR-24, février 1993.
- [xxi] PAULK M. C., WEBER C., GARCIA S. M., CHRUSIS M. B. & BUSH M., *Key Practices of the Capability Maturity Model, Version 1.1*, SEI, CMU/SEI-93-TR-25, février 1993.

- [xxii] PAULK M., et al., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, 1995.
- [xxiii] RENAULT A., *Etude critique d'une utilisation du modèle SPICE*, OWPL-FUNDP, juillet 1998.
- [xxiv] RENAULT A., *Micro-évaluation des Pratique Logicielles - Etude de cas*, OWPL-FUNDP, janvier 1999.
- [xxv] SEI, *Benefits of CMM-Based Software Process Improvement : Initial Results*, CMU/SEI-94-TR-013, ESC-TR-94-013, Août 1994.
- [xxvi] SEI, *Process Tailoring and the Software Capability Maturity Model, Technical Report*, CMU/SEI-94-TR-024, ESC-TR-94-024, Novembre 1995.
- [xxvii] ZAHRAN S., *Software Process Improvement : Practical Guidelines for Business Successes*, Addison-Wesley, 1997.
- [xxviii] ZUBROW D., et. al., *Maturity Questionnaire*, SEI, CMU/SEI-94-SR-007, Avril, 1994.

## 7. Annex I: Mapping OWPL – ISO/IEC12207:1995 –AMD1 & AMD2

Proc_Id	OWPL Process	Prac_Id	OWPL Practice	ID	12207:AMD1&2 Process	ID	12207:1995 Activity
REQU	Requirements Management	REQU/PR01/03	Analyzing Requirements	F3.1.1/ F1.3.4	Requirements elicitation/ Software requirements analysis	5.3.4	Software requirements analysis
REQU	Requirements Management	REQU/PR02/03	Managing change Requests	F2.11	Change request management		<i>Not Explicitly Adressed</i>
REQU	Requirements Management	REQU/PR03/03	Validating the software product	F2.5	Validation Process	6.5.2	Validation
DOCS	Documentation	DOCS/PR01/03	Identifying documentation requirements	F2.1	Documentation Process	6.1.1	Process implementation
DOCS	Documentation	DOCS/PR02/03	Developing documentation requirements	F2.1	Documentation Process	6.1.3	Production
DOCS	Documentation	DOCS/PR03/03	Providing the documentation documents to the appropriate people		<i>Not Explicitly Adressed</i>		<i>Not Explicitly Adressed</i>
PLAN	Project Planning	PLAN/PR01/06	Decomposing the project in tasks	F3.1.3	Project management	7.1.2	Planning
PLAN	Project Planning	PLAN/PR02/06	Identifying constraints associated to a task	F3.1.3	Project management	7.1.2	Planning
PLAN	Project Planning	PLAN/PR03/06	Estimating the necessary resources	F3.1.3	Project management	7.1.2	Planning
PLAN	Project Planning	PLAN/PR04/06	Analyzing risks	F3.1.5	Risk management		
PLAN	Project Planning	PLAN/PR05/06	Creating the planning	F3.1.3	Project management	7.1.2	Planning
PLAN	Project Planning	PLAN/PR06/06	Adapting the planning	F3.1.3	Project management	7.1.3	Execution and control
TROV	Project Tracking and Oversight	TROV/PR01/04	Recording project progress	F3.1.3	Project management	7.1.3	Execution and control
TROV	Project Tracking and Oversight	TROV/PR02/04	Analyzing project progress	F3.1.3	Project management	7.1.3	Execution and control
TROV	Project Tracking and Oversight	TROV/PR03/04	Taking corrective decisions	F3.1.3	Project management	7.1.3	Execution and control
TROV	Project Tracking and Oversight	TROV/PR04/04	Conducting project assessment and wrap-up meeting	F3.1.3	Project management	7.1.5	Closure
DVLP	Development	DVLP/PR01/05	Functional analysis	F1.3.5	Software design	5.3.5	Software architectural design
DVLP	Development	DVLP/PR02/05	Conception	F1.3.5	Software design	5.3.6	Software detailed design
DVLP	Development	DVLP/PR03/05	Coding	F1.3.6	Software construction	5.3.7	Software coding and testing
DVLP	Development	DVLP/PR04/05	Debugging	F1.3.6	Software construction	5.3.7	Software coding and testing
DVLP	Development	DVLP/PR05/05	Deployment and corrective maintenance	F1.3.11	Software installation	5.3.12/ 5.3.13	Software installation/Software acceptance support
TEST	Tests	TEST/PR01/04	Defining a test strategy		<i>Not Explicitly Adressed</i>		<i>Not Explicitly Adressed</i>
TEST	Tests	TEST/PR02/04	Creating tests suites	F1.3.8	Software testing	5.3.8	Software integration
TEST	Tests	TEST/PR03/04	Tracking and overseeing test	F1.3.8	Software testing	5.3.9	Software qualification testing
TEST	Tests	TEST/PR04/04	Conducting tests	F1.3.8	Software testing	5.3.9	Software qualification testing
CONF	Configuration Management	CONF/PR01/03	Identifying work products	F2.2	Configuration Management Process	6.2.2	Configuration identification
CONF	Configuration Management	CONF/PR02/03	Identifying relationships between work products	F2.2	Configuration Management Process	6.2.3	Configuration control
CONF	Configuration Management	CONF/PR03/03	Tracking the configuration	F2.2	Configuration Management Process	6.2.4	Configuration status accounting

Proc_Id	OWPL Process	Prac_Id	OWPL Practice	ID	12207:AMD1&2 Process	ID	12207:1995 Activity
CONF	Configuration Management	CONF/PR01/03	Identifying work products	F2.2	Configuration Management Process	6.2.2	Configuration identification
CONF	Configuration Management	CONF/PR02/03	Identifying relationships between work products	F2.2	Configuration Management Process	6.2.3	Configuration control
CONF	Configuration Management	CONF/PR03/03	Tracking the configuration	F2.2	Configuration Management Process	6.2.4	Configuration status accounting
SUMA	Subcontract Management	SUMA/PR01/04	Writing the invitation to tender	F1.1.1	Acquisition Preparation	5.1.2	Request-for-proposal [-tender] preparation
SUMA	Subcontract Management	SUMA/PR02/04	Selecting subcontractors	F1.1.2	Supplier selection	5.1.3	Contract preparation and update
SUMA	Subcontract Management	SUMA/PR03/04	Tracking subcontract agreement (and requirement evolution)	F1.1.3	Supplier monitoring	5.1.4	Supplier monitoring
SUMA	Subcontract Management	SUMA/PR04/04	Receiving subcontract work	F1.1.4	Customer acceptance	5.1.5	Acceptance and completion
QUAL	Quality Assurance	QUAL/PR01/05	Defining and adapting standards, norms and procedures	F3.1.4	Quality management		<i>Not Explicitely Adressed</i>
QUAL	Quality Assurance	QUAL/PR02/05	Monitoring product quality	F2.3	Quality Assurance Process	6.3.2	Product assurance
QUAL	Quality Assurance	QUAL/PR03/05	Monitoring processes quality	F2.3	Quality Assurance Process	6.3.3	Process assurance
QUAL	Quality Assurance	QUAL/PR04/05	Taking corrective decisions	F2.3	Quality Assurance Process	6.3.4	Assurance of quality system
QUAL	Quality Assurance	QUAL/PR05/05	Evaluating work and team members	F3.4.1	Human resource management		<i>Not Explicitely Adressed</i>
CPTL	Experience Capitalization	CPTL/PR01/03	Analyzing former projects	F3.3.3	Process improvement	7.3.3	Process improvement
CPTL	Experience Capitalization	CPTL/PR01/03	Tailoring the training framework	F2.3	Quality Assurance Process	6.3.1	Process implementation
CPTL	Experience Capitalization	CPTL/PR01/03	Spreading continuously information on quality manual		<i>Not Explicitely Adressed</i>		<i>Not Explicitely Adressed</i>