

Experience Management for Very Small Entities: Improving the Copy-paste Model

Vincent Ribaud, Philippe Saliou

Département d'Informatique
Université de Bretagne Occidentale, UEB
20 avenue Le Gorgeu, CS 93837
29238 Brest Cedex, France
{Vincent.Ribaud, Philippe.Saliou}@univ-brest.fr

Claude Y. Laporte

Département de génie logiciel et des TI
École de technologie supérieure
1100 rue Notre-Dame Ouest
Montréal (Québec), Canada, H3C 1K3
Claude.Y.Laporte@etsmtl.ca

Abstract— The emerging ISO/IEC 29110 standard Lifecycle profiles for Very Small Entities is developing a “Generic” profile group applicable to a vast majority of very small entities (enterprises, organizations, departments or projects) having up to 25 people, that do not develop critical software and have typical situational factors. The developers of the standard, ISO/IEC JCT1/SC7 Working Group 24, recommended the use of pilot projects as a mean to trial the adoption of the new International standard in small organizations. Accordingly an ISO/IEC 29110 pilot project has been established between the Software Engineering group of Brest University and a 14-person company with the aim of establishing an engineering discipline for a new web-based project. As the project proceeded, it became apparent that the current set of ISO/IEC 29110 documents describing a first profile, the Basic profile, was not sufficient to sustain this VSE in its SE activities. What was needed was to organize the knowledge contained in them. The results of this pilot study are providing VSEs with a simple Experience Management system which is compatible with the emerging ISO/IEC 29110 standard. It is founded on two principles: 1) keeping the Content Management System-based Experience Management infrastructure as simple as possible, structured with the decomposition of the ISO/IEC 29110 processes; and 2) the requirement of Experience Management dedicated processes, taken from D. Schön’s work on the reflection-on-action approach to learning.

Keywords: *software engineering processes, experience management, very small entities, ISO standards.*

I. INTRODUCTION

Very Small Entities (VSEs) are enterprises, departments or projects having up to 25 people [1]. VSEs are recognized as being very important to the software economy, producing software components either as stand-alone products or as elements to be integrated into large software systems. Many international standards and models have been developed to capture proven engineering practices. However, they were not designed having in mind the needs and expertise of VSEs. They are consequently difficult to apply in such settings [1]. The ISO/IEC 29110 software engineering standard - Lifecycle Profiles for Very Small Entities (VSEs) [2] is under development since 2005 to address these difficulties, and is specifically tailored to the needs of VSEs. Profiles are subsets of appropriate standards elements, relevant to the VSE context, for example, processes and

products of main software engineering (SE) standards. Because software engineers in a VSE are continuously using SE processes and producing SE products in different projects, a significant help could be expected from an Experience Management (EM) system for a VSE that will provide a way to relate and integrate those projects experiences.

One of the main ideas defended in this paper is that an EM system for a VSE should be constructed on a framework suitable for that entity, but derived from a standardized Process Reference Model (partly presented in Section III.C) taken from the emerging ISO/IEC 29110 standard [2].

EM solutions to organizing knowledge can be supported by experience factories (EF) [3]. “*EM includes methods, techniques, and tools for identifying, collecting, documenting, packaging, storing, generalizing, reusing, adapting, and evaluating experience; and for the development, improvement, and execution of all knowledge-related processes [4].*” EF is defined as “*an infrastructure designed to support experience management*” and “*supports the collection, preprocessing, and dissemination of experiences*” [4]. This paper presents a simple knowledge management system intended to gather, link, and reuse knowledge about SE activities. Requirement Analysis and its associated work products will be used as an example.

Professional competency management is focused on the development of professional skills and attitude. These components are usually addressed in a ‘practicum’ or ‘clinical work’, and the concept of reflection, inspired by D. Schön’s work [5], is central to this competency development. The knowledge management system was designed based on two main guiding principles: the extraction of knowledge of existing SE standards – providing the system with a bootstrap, and favoring the building of new knowledge by the software engineers themselves – a process required to maintain accurate and ‘living’ knowledge.

The next section provides an overview of ISO/IEC 29110 standard, EM and EF, and related work. The standard is discussed in Section III, and a case study introduced focused on requirements analysis and tests activities. In Section IV, we present our work on EM for a VSE and discuss some facts of the case study. We conclude the paper with the brief statement of a few perspectives.

II. REQUIREMENTS AND RELATED WORK

In this section, we present the ISO/IEC 29110 initiative, draft what could an Experience Factory be and overview D. Schön's reflection-on-action work. Related work is also discussed.

A. SE Standards for Very Small Entities

1) ISO Terminology

A Base Standard is an approved International Standard or ITU-T Recommendation [6]. An International Standardized Profile (IS) is a harmonized document on which there is international agreement, and which describes one or more profiles [6]. A Profile is a set of one or more base standards and/or ISs, including, where applicable, the identification of chosen classes, conforming subsets, options, and parameters of those base standards, or ISs, required to accomplish a particular function [6]. A Technical Report (TR) is developed like a standard, but its purpose is simply to provide technical information, and not to make requirements on implementation. A TR is also cost-free.

2) ISO Initiative

SE standards and methods often neglect the needs and problems of small and medium-sized organizations, which constitute a major part of the software industry. In 2005, the ISO recognized the needs and problems of VSEs and established a Working Group (WG24) mandated to develop a set of standards and technical reports suitable for these entities. The emerging ISO/IEC 29110 standard constitutes a set of guidelines for use by very small software entities. The guides are based on subsets of appropriate standards elements, referred to as VSE profiles [2], relevant to the VSE context, for example, processes and outcomes of ISO/IEC 12207 [7] and products of ISO/IEC 15289 [8].

The Generic profile group is targeted at VSEs that do not develop critical software and have typical situational factors. The Generic profile group is composed of 4 profiles: Entry, Basic, Intermediate, and Advanced. The Basic Profile [2] applies to VSEs involved in the development of a single software application by a single project team with no special risk or situational factors. By design, it excludes many of the ISO/IEC 12207 processes.

The standard is composed of several parts. As specified in [2], Parts 1 and 5 target VSEs, Part 3 targets assessors and VSEs, and Parts 2 and 4 target standards producers, tool vendors, and methodology vendors. If a new profile is needed, Parts 4 and 5 can be further developed without impacting existing documents, and they would become Part 4-x and Part 5-x-y respectively through the ISO/IEC standardization process.

The simplest path for a VSE is to start with Part 5-1-2: Management and Engineering Guide – Basic Profile. Using the Guide, a VSE can benefit in the following ways [9]:

- An agreed set of project requirements and expected products are delivered to the customer;
- A disciplined management process, which provides project visibility and corrective actions for project problems and deviations, is performed;

- A systematic software implementation process, which satisfies customer needs and ensures quality products, is followed.

3) Deployment Packages

Once the ISO/IEC TR 29110-5-1-2 has been downloaded, at no cost, from the ISO website, a VSE may consider that the help provided in it is insufficient to guide the implementation. Deployment Packages (DPs), by contrast, can be expected to provide significant help. We defined a DP as “a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE” [11]. The elements of a typical DP are: process description (activities, inputs, outputs, and roles), a guide, a template, a checklist, an example, presentation material, references and mapping to standards and models, and a list of tools [11]. The mapping is given only as information to show that a DP has explicit links to standards, such as ISO/IEC 12207, or to models, such as the CMMI. So, by deploying and implementing the package, a VSE can visualize the concrete steps required to achieve or demonstrate coverage. Packages are designed so that a VSE can implement their content without having to implement the complete framework at the same time.

4) Pilot Projects

Pilot projects are an important means for reducing risks and learning more about the organizational and technical issues associated with the deployment of SE practices. A successful pilot project is also an effective means for encouraging the adoption of new practices by members of a VSE [11]. DPs are intended to apply the ISO/IEC 29110 standard in a VSE. Tailoring software processes to a VSE constitutes a kind of process improvement. A pilot project may also be an initial implementation of a DP, which provides WG24 with feedback of the improvement proposals before the DP is adopted as a standard.

B. Experience Factory

In order to achieve experience reuse, Basili et al. [12] have proposed an organizational framework that separates project-specific activities from reuse packaging activities, with process models to support each activity.

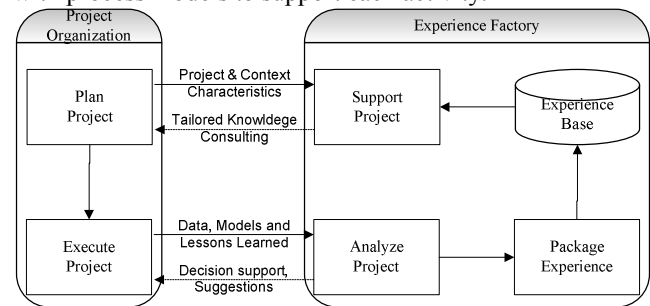


Figure 1. Experience factory (adapted from Ras et al. [4]).

The framework, represented in Figure 1, defines two separate organizations: a project organization, intended to deliver the system required by the customer, and an Experience Factory (EF), the role of which is to monitor and analyze project developments, develop and package

experience for reuse in the form of knowledge, processes, tools, and products, and supply them to the project organization upon request [12].

Separating the project from the experience organization, physically or logically, may relieve project teams of the tasks required by EM, but is, in our opinion, inapplicable in very small entities, even in small and medium-sized software companies (up to 250 employees). We agree with [13] that small and medium-sized software organizations (SMEs) need a more lightweight means of creating these knowledge bases with minimal overhead. Wiki-based repositories are often used as knowledge repositories, because the wiki concept easily integrates users into the knowledge-sharing process in SMEs [14].

C. The Reflective Practitioner

In many EF implementations, effort is put into extracting knowledge from valid experiences. Extracted knowledge facilitates experience reuse and learning. In Figure 1, adapted from [4], arrows from left (Project Organization) to right (Experience Factory) indicate knowledge extraction. Argyris and Schön explored the nature of organizational learning and defined two kinds of learning: simple-loop and double-loop [15]. Knowledge transfer is a double-loop learning activity. We dashed the right-to-left arrows (knowledge transfer), because it is a much harder issue for engineers to cope with, especially novice engineers.

D. Schön [16] argues that experienced professionals deal with the ‘messiness’ of practice not by consulting the research knowledge base, but by engaging in what he calls ‘reflection-in-action’, which is sometimes described as ‘thinking on our feet’. The notion of the repertoire is very important in Schön’s approach. Practitioners build up a collection of ideas, examples, situations, and actions. “A practitioner’s repertoire includes the whole of his experience insofar as it is accessible to him [her] for understanding and action [5, p. 138].” We believe that there can be a considerable gap between the structure of an engineer’s repertoire (and hence the project organization that he or she may use) and the structure of the EF. As the most common knowledge pattern transfer is the copy-paste model, it may be that the transferred knowledge does not make sense to the user, especially novices with an almost empty repertoire.

Ras et al. [17] address this problem with an approach called ‘learning space generation’, which enriches experience packages with additional information from specifications provided either by the instructor or by the student. The learning space is presented by means of Wiki pages within a specialized Wiki based on the Software Organization Platform (SOP). Our approach does the opposite. Rather than providing engineers with access to the experience packages, we essentially provide task description exemplars and product exemplars created in small projects.

III. A STANDARDIZED PROCESS PROFILE FOR VSEs

In this section, we present the context of the pilot project, expectations of the VSE, and the application of the emerging ISO/IEC 29110 standard to this project.

A. Implementation of Standardized Processes

At the core of the ISO/IEC 29110 standard is a Management and Engineering Guide (ISO/IEC DTR 29110-5) [9] focusing on Project Management and Software Implementation and an Assessment Guide (ISO/IEC DTR 29110-3) [19]. ISO/IEC DTR 29110-5 provides a practical guide related to the ISO/IEC FDIS 29110-4-1 standard [10] identified as a Basic Profile. For instance, the starting point for ISO/IEC 29110 use for requirements is the SI.2 Software Requirements Analysis activity, its list of tasks: SI.2.1 to SI.2.7, and the associated roles.

Deployment Packages (DP) provide VSEs with assistance in adopting standards through a DP Repository <http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html>. For instance, DP Software Requirement Analysis [21] simplifies task decomposition and provides a step-by-step method for each task.

B. Pilot Project

1) Requirements

Fenton et al. state in [18]: “For 25 years, software researchers have proposed improving software development and maintenance with new practices whose effectiveness is rarely, if ever, backed up by hard evidence.” They suggest several ways to address this problem, in particular to carefully design and measure experiments, such as pilot projects.

2) Context of the VSE

A VSE with a staff of 13 (2 of them software engineers) requested the help of our SE group in the spring of 2009. The VSE designs, builds, develops, and sells counting systems intended to collect and analyze data on visits to public or private sites. Initially designed for counting pedestrians, this VSE’s products now cover bikes, horses, and cars. Counting systems are based on stand-alone counter boxes (including sensors, a power supply, data storage, and data exchange) and a software able to collect, analyze, present, and report counting data. In the previous software chain, the set of data was downloaded from counters by infrared link or GSM, was stored on personal computers, and then was transmitted via a file transfer utility.

3) The new Software Project

Because of its clients’ requirements and the products supplied by the competition, the VSE began a complete reconstruction of its software chain in order to transform it into a Web-based system, called Eco-Visio, intended to host the data of fleets of counting systems for each client and capable of processing statistics and generating analytical reports on counting. At the end of June 2009, the VSE hired a graduate of Brest University who had done his final internship at the VSE. At the same time, we visited the VSE and initiated a pilot project with the intention of transferring a part of the ISO/IEC 29110 standard to the VSE and its specific context. Project stakeholders decided to focus on two SE activities: 1) establishment of a practical technique for gathering and managing requirements; and 2) improvement of the system’s reliability with a disciplined test process.

The new software project, completed at the end of March 2010, was released as the first version of the new Eco-Visio Web-based system.

C. Basic Profile

1) Basic Profile Processes

The Generic Profile Group [10] is a collection of four profiles (Entry, Basic Intermediate, Advanced), providing a progressive approach to satisfying a vast majority of VSEs that do not develop critical software and have typical situational factors. The Basic Profile applies to a VSE that is involved in software development of a single application by a single project team with no special risk or situational factors. The objective of the project is to fulfill an external or internal contract. The internal contract between the project team and their client need not be explicit.

The Basic Profile is made up of two processes: Project Management (PM) and Software Implementation (SI). A process is defined as “a set of interrelated or interacting activities which transforms inputs into outputs [7].” Table I provides the process/activity breakdown, and presents tasks related to requirements and tests (which are the focus of the pilot project cited above).

TABLE I. BASIC PROFILE PROCESS BREAKDOWN

Process	Activities	Pilot project related tasks
PM Project Management	PM.1 Project Planning	PM.1.1
	PM.2 Project Plan Execution	PM.1.13, PM.1.14
	PM.3 Project Assessment and Control	PM.2.2 and PM.2.4
	PM.4 Project Closure	PM.3.5
SI Software Implementation	SI.1 SW Implementation Initiation	-
	SI.2 SW Requirements Analysis	SI.2.2, SI.2.3, SI.2.4
	SI.3 SW Architectural and Detailed Design	SI.3.5, SI.3.6
	SI.4 SW Construction	SI.4.4
	SI.5 SW Integration and Tests	SI.5.4
	SI.6 Product Delivery	-

ISO/IEC DTR 29110-5-1-2 [9] is intended to guide the Basic Profile implementation of PM and SI processes described in ISO/IEC FDIS 29110-4-1 [10]. These processes integrate practices based on the selection of ISO/IEC 12207 SW life cycle processes and ISO/IEC 15289 information product (documentation) standards elements. DPs will facilitate the implementation of these processes.

2) Basic Profile Products

Clause 9 of ISO/IEC FDIS 29110-4-1 [10], establishes the normative list of Basic Profile work product and deliverable specifications. There are 23 work products, which can be input, output, or internal products of processes, activities, or tasks.

3) Process Assessment

ISO DTR 29110-3 [19] is an Assessment Guide applicable to all VSE profiles. It is compatible with ISO/IEC 15504-2 and ISO/IEC 15504-3. The assessment has two purposes: 1) to evaluate process capability based on a two-dimensional assessment model (from the ISO/IEC

15504:2006 standard [20]); and 2) to determine whether or not an organization achieves the targeted VSE Profile based on the process capabilities evaluated [19]. A VSE-specific Process Assessment Model (PAM) can be derived by selecting only the assessment indicators in the ISO/IEC 15504-5 ‘An Exemplar PAM’ relevant to the corresponding process outcomes, as defined in ISO/IEC FDIS 29110-4-1.

4) Performing the ISO/IEC 29110 Requirements Analysis

ISO/IEC FDIS 29110-4-1 provides a set of cohesive tasks for each activity. Also established here are the VSE needs and suggested competencies. For instance, it defines the SI.O2 objective: “Software requirements are defined, analyzed for correctness and testability, approved by the Customer, baselined and communicated. Changes to them are evaluated for cost, schedule and technical impact previously to be processed [10, p. 8].”

ISO/IEC DTR 29110-5-1-2 details, for each activity of the PM and SI processes, the tasks to be performed: role, description of the task, and input and output products. For instance, it defines tasks SI.2.1 to SI.2.7 and their associated output products: Requirements Specification, Verification Results, Change Request, Validation Results, and Software User Documentation.

DP Software Requirement Analysis [21] simplifies task decomposition: requirement identification, requirement refinement and analysis, requirement verification and validation, and requirement change management. A step-by-step method is described for each of these four tasks. The DP also provides a Software Requirement Specification template. Training materials and an Excel-based traceability tool can be downloaded from the publicly accessible WG24 website. Despite all this helpful material, the young VSE engineer was not able to complete Requirements Analysis.

5) Problem Analysis

Based on feedback from a VSE employee that he was not able to perform the SI.2 Software Requirements Analysis activity, the authors analyzed the problem.

Action theory studies what an actor does, in a given situation, in order to achieve objectives. Argyris and Schön [22] made a distinction between *espoused* theories, those that an individual claims to follow, and *theories-in-use*, those that can be inferred from action. Espoused theory and theory-in-use may be inconsistent, and the agent may or may not be aware of any inconsistency. By definition, the agent is aware of espoused theory. Theories-in-use can be made explicit by reflecting on action [23].

Software companies use SE and software quality standards as the foundation of their quality assurance process or of their quality management system. As software companies claim to follow and respect standards, we may think that these standards constitute a part of espoused theories of software engineers, especially Process Assessment and Process Reference Models. In the software field, we observe that a software engineer may have a work behavior – her/his theories-in-use – which often runs quite contrary to the organizations’ processes, practices, and

procedures that she/he is supposed to follow and speak about, i.e., espoused theories.

What happened to that young engineer? Through ISs, TRs, DP, and other materials, he received a great deal of information on espoused theory. However, as his repertoire of experience (and VSE Experience Repository) was all but empty, he could not act in accordance with any theory-in-use.

Managing experience in a repository may provide VSE engineers with a simple form of knowledge management. But, as we will see in the next section, an Experience Repository requires supplementary processes in order to support knowledge transfer.

IV. EXPERIENCE MANAGEMENT FOR THE VSE

Chan and Chao present a research survey conducted among 68 small and medium-sized enterprises (SME) which have implemented Knowledge Management (KM) initiatives [24]. SMEs are quite bigger than targeted VSEs but lessons learned in this survey apply also to VSEs. Effective KM is influenced by two types of KM capability, infrastructure and process, which have to be deployed. This section presents a simple Content Management System (CMS)-based infrastructure to manage experience and some activities that may be part of EM processes.

A. An Experience Repository

1) Related work

A significant part of EM in a software company should be about software and documentation reuse (code reuse is outside the scope of this paper). So, the primary inputs of our system are documentation deliverables: plans, requirements, design specifications, data schemas, test cases, and so forth. Publishing and content management systems (CMS) are generally used as the basis for a documentation management infrastructure. But several authors have criticized the rigidity of the editorial control required by a CMS [25] and the need to balance structure/constraint and flexibility [26]. Some are promoting the use of Wikis and RDF (Resource Description Framework) to resolve these issues [27].

Wikis are probably a suitable tool for facilitating collaborative design and development, and may be viewed as part of the project repository (see Figure 1), but requirements for an EM infrastructure are different. Rech et al. identified several challenges related to knowledge transfer and management processes for SMEs in the software sector: recording, reusing, locating, and sharing information [14]. The authors evaluated a small software enterprise and a micro software enterprise with reuse policies in place. They point out that the engineers have little confidence in knowledge transfer, because only a few people have created reusable documents. They also note that the workflow for reusing knowledge is slow and typically demotivating, because multiple sources have to be searched manually and documents belonging together weren't grouped together or linked [14]. As we will see in the next section, a CMS-based system with a simple and fixed structure may resolve most of these issues.

2) Experience Repository Infrastructure

Managing an Experience Repository for a small project may be greatly facilitated if the structure is kept as simple as possible. We should also avoid a plethora of artifacts. Our proposal is that, whenever a project is completed, the project closure activity create its own space in the CMS and use the Process/Activity decomposition of ISO/IEC FDIS 29110-4-1 [10, Clause 7] as the structure for that space. Then, only the main deliverables of the project, as they are defined in ISO/IEC DTR 29110-5-1-2 [9, Clause 4], will be stored in the right place in the structure.

Table II shows the structure and content of the Experience Repository for some representative activities of each process. In order to illustrate our work, we added the activity-related tasks. The left column provides links to the ISO/IEC 12207 activities profiled in the mentioned parts of ISO/IEC 29110. We added in italics a proposal (submitted in [28]) for the management of deliverables related to support activities.

TABLE II. STRUCTURE AND CONTENT OF THE EXPERIENCE REPOSITORY

12207	Activity	Tasks	Output products
Project Management Process			
6.3.1.3.3, 6.3.2.3.1, 6.3.2.3.2	PM.2 Project Plan Execution	<ul style="list-style-type: none"> • PM.2.1 Review Project Plan • PM.2.2 Change request analysis • PM.2.3 External revision meeting • PM.2.4 Internal revision meetings 	Project Plan Change Request Meeting Record
...			
Software Implementation Process			
6.4.1.3.1, 6.4.1.3.2, 6.4.1.3.3, 6.4.1.3.4, 6.4.1.3.5, 7.1.2.3.1	SI.2 SW Require- ments Analysis	<ul style="list-style-type: none"> • SI.2.2 Document requirements • SI.2.3 and 2.4 V & V requirements 	Requirements Specifications V & V Results
7.1.3.3.1, 7.1.4.3.1	SI.3 SW architectural and detailed design	<ul style="list-style-type: none"> • SI.3.3 Document software design • SI.3.4 Software design verification 	Software Design Traceability Record Verification Results
...			
Management and Implementation Support Process			
6.2.1.3.1, 6.2.1.3.3	Method and tool support	<ul style="list-style-type: none"> • <i>Process establishment</i> • <i>Process improvement</i> • <i>Tool support</i> 	<i>Process implementation recommendations</i> <i>Tool usage guide</i>

The infrastructure is not intended to be a project repository (the left part of Figure 1) hosting project deliverables in different versions. The infrastructure forms part of an experience repository intended to record the final state of the project and to provide further projects with exemplars of deliverables. The Alfresco content platform (<http://www.alfresco.com/products/wcm>) is used as a Web content management suite, mainly for providing an upload-download system organized into a hierarchy of space, with

the possibility of a fine-grained control of users' rights over spaces. As mentioned above, the space hierarchy structure is, for each project, the Process/Activity decomposition of ISO/IEC FDIS 29110-4-1 [10]. Each space hosts a variety of work products of ISO/IEC DTR 29110-5-1-2 [9]. Examples of these products are given in Table II, column 4.

B. KM Support Processes

According to [29], knowledge can be created through dedicated acquisition, conversion, application, and protection of knowledge assets. In the survey of 68 SMEs by Chan and Chao [24], most of the respondents stated that they encounter knowledge capture problems related to time, place, and people. We decided to provide the VSE with two levels of Experience Management Process: a copy-paste level, and a continuous tailoring level.

1) Copy-paste Activities

The copy-paste process is designed to be as simple as possible. Clauses 4.2.8 and 4.3.8 of ISO/IEC DTR 29110-5-1-2 [9] proposes task decomposition of the PM and SI processes for each activity, together with inputs and outputs of each task. So, we can establish the workflow for each of the 23 work products (cf. §III.C.2). For instance, Figure 2 presents the workflow of Work Product 11, *Requirements Specification*.



Figure 2. WP11 *Requirements Specification* workflow

It may happen that a work product workflow spans several activities of the same process, such as WP17 *Software User Documentation*, which covers SI.2 to SI.5, and even PM and SI processes, such as WP8 *Project Plan*, which covers PM.1 to SI.6.

The VSE needs a simple model to locate, store, and retrieve work products, according to the Process Reference Model used. Our proposal is to locate a work product inside the CMS space associated with the last activity that outputs the final version of this work product. So, WP11 *Requirements Specification* will be located in the 'SI.2 SW Requirements Analysis' space, while WP17 *Software User Documentation* will be located in the 'SI.5 SW Integration and Tests' space and WP8 *Project Plan* in the 'SI.6 Product Delivery' space.

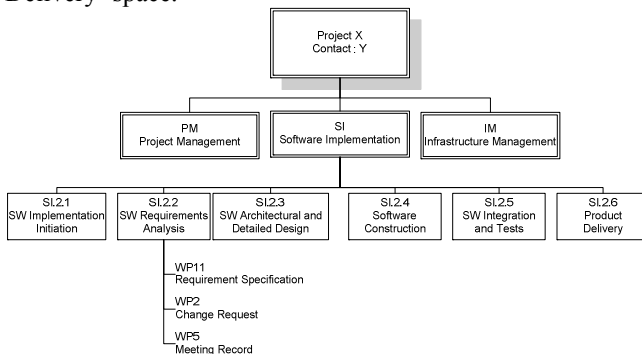


Figure 3. Structure and content of a project in the Experience Repository.

The task of storing work products in an Experience Repository space associated with the project will be allocated to the PM.4 Project Closure activity, which is the responsibility of the Project Manager (PM) role. With this simple copy-paste EM process, the PM role stores artifacts, and (ideally) each VSE employee can access and copy the artifacts of her/his choice. An extract from the structure and content of a project space is given in Figure 3.

2) Reflection-on-action

To take up the challenges of their practice, practitioners rely on their repertoire of experience, along with a certain ingeniousness acquired during that practice, rather than on knowledge-oriented curricula or formulae learned during their basic education. D. Schön describes this repertoire as follows: *"The practitioner has built up a repertoire of ideas, examples, situations and actions. [...] When a practitioner makes sense of a situation he perceives to be unique, he sees it as something already present in his repertoire. To see this site as that one is not to subsume the first under a familiar category or rule. It is [...] to see the unfamiliar, unique situation as both similar to and different from the familiar one, without at first being able to say similar or different with respect to what. The familiar situation functions as a precedent, or a metaphor, or [...] an exemplar for the unfamiliar one"* [5, p.138].

In order to help VSE employees fill their own repertoire, and consequently the VSE Experience Repository, we have designed practices that may help software engineers to become 'reflective' practitioners. These practices are generally borrowed from two streams: industrial – process improvement and product assessment – and Schön's reflection-on-action theory.

In order to bootstrap the Experience Repertory for a given activity in SE (e.g., requirements analysis or design), we have developed an approach based on the tailoring of an activity before performing the activity itself. This approach has been presented through the specific case of the design in [30].

The approach is generally implemented in two steps: 1) tailoring the activity to acquire a minimal structure through a deductive approach (by writing a guide, for instance); and 2) initializing the repertory through an inductive approach, with the use of retro-engineering, for instance. This approach is a pragmatic answer to the lack of support and training that may be experienced in small projects, where the main effort is concentrated on project management and software development tasks.

3) Training Packages

If we want train to an SE activity, our proposal is to organize the engineer's training path through small units of work, called training tasks. The description of the task is designed as a theater scene: the scene is the reference context where action takes place; it aims to maintain unity of place, time, and action, and it is a site where a situation can occur and where people perform actions (and learn). It also serves as a location for action scenarios, for role distribution, and for mobilizing resources and means. The various components of a scene, along with their linkages, are

depicted on a training card. The card structure is standardized:

- Related 29110 Process/Activity

This reference (for instance, SI/SI.2 SW Requirements Analysis) provides a smooth link to ISO/IEC 29110, and through the profile to ISO/IEC 12207 and 15504.

- Role

Role (for instance, Analyst) is a quick reference to the ISO/IEC 29110.

- Task title and objectives

Similar to Process Title, Process Purpose, and Process Outcomes, as defined in ISO/IEC 12207.

- Step-by-step guide

A comprehensive description of the work to be done, intended to be a practical guide to completing the task.

- Resources

The set of resources required. It may include the hosting of technical support (such as Oracle Metalink) that a technology transfer center is able to afford when the cost is out of reach for a VSE.

- Output products

Generally a methodological survey, a tool usage guide, or an installation manual.

The set of training activities that a VSE engineer should perform is incorporated into a Training Package (TP) (analogous to the ISO/IEC 29110 Deployment Package). Developing the concept of the TP is outside the scope of this paper, but suffice to say that a TP is primarily intended to provide self-training on SE activities, with the supplementary goal of initiating and developing a strategy of capitalizing on this knowledge and transferring it to VSE employees.

4) Empirical Evaluation

The VSE engineer mentioned in § II.A.3) was provided with two TPs on Requirements at the end of 2009. The first was intended to achieve a basic maturity level on ISO/IEC 29110 Requirements Management (through the study of an SI.2 activity and a review of a ‘real’ WP11 *Requirements Specification*), and the second involved performing a Requirements Analysis on a ‘real’ case. The first package was made up of 3 training scenes and the second of a single one. Each TP was calibrated to a week of self-training. The VSE engineer worked through both packages in January 2010.

Favoring reflection-in-action through TPs is, in our opinion, a kind of software improvement. Although no measurements are easily defined and performed to confirm this, there is empirical evidence of it in the form of ‘customer’ satisfaction.

The VSE engineer reported that he was now ready to apply the SI.2 SW Requirements Analysis to the Eco-Visio project. As the specifications were established by a subcontractor, he merely reviewed the existing Requirements Specification and rewrote parts of it in order to verify conformity with the template provided in the DP, Software Requirement Analysis [21]. Once updated, the WP11 *Requirement Specification [Validated]* served as an input to SI.5, SW Integration and Tests. The system has been deployed since the end of March 2010, and load testing and

application optimization should soon be completed. Then, defects will have to be corrected through a short cycle of SI activities.

As an empirical measure of satisfaction with the approach, the VSE asked for a similar approach for SI.5 SW Integration and Tests. In particular, the VSE wanted assistance in establishing a disciplined Change Request Process. This TP is under construction, and we should begin with the DP, Software Testing [31], as a basis for the whole TP. Probably because tests occur in many SE activities, this DP is organized in a manner that spans PM and SI tasks, and raises many new questions.

Figure 4 shows all the Infrastructure and Process issues that we have addressed in this section.

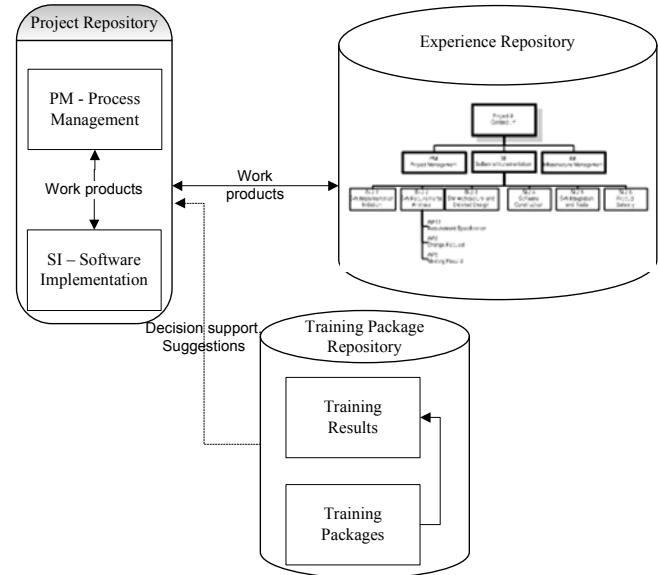


Figure 4. Overview of the EM Infrastructure and Process.

V. CONCLUSION AND FUTURE WORK

We propose to provide a VSE with a simple Experience Management system, compatible with the emerging ISO/IEC 29110 standard. Two hypotheses are posed: (1) the EM infrastructure is kept as simple as possible with the use of a CMS structured with the decomposition of the PM and SI processes; (2) EM requires dedicated processes that may be taken from D. Schön’s reflection-on-action work. As a case study, the needs of a VSE and the solutions that we provided are reported.

Further work is required to consider how the concept of Training Packages could complement that of Deployment Packages.

REFERENCES

- [1] C. Laporte, “The Development of International Standards for Very Small Entities: Historical Perspectives, Achievements and Way Forward” Joint International Council on Systems Engineering (INCOSE) / Concordia Institute for Information Systems Engineering (CIISE) Distinguished Seminar, 2010.
- [2] ISO/IEC DTR 29110-1:2010, Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE) -- Part 1: VSE profiles

Overview, Geneva: International Organization for Standardization (ISO), 2010.

- [3] V. Basili, G. Caldiera, and D. Rombach, "Experience Factory," in Encyclopedia of SE, vol. 1, Hoboken: Wiley, 2002, pp. 476-96.
- [4] E. Ras, R. Carbon., D. Decker, and J. Rech, "Experience Management Wikis for Reflective Practice in Software Capstone Projects," IEEE Transactions on Education, vol. 50 (4), Nov. 2007, pp. 312-320.
- [5] D. Schön, The Reflective Practitioner, New York: Basic Books, 1983.
- [6] ISO/IEC TR 10000-1:1998, Information technology -- Framework and taxonomy of International Standardized Profiles -- Part 1: General principles and documentation framework, Geneva: International Organization for Standardization (ISO), 1998.
- [7] ISO/IEC 12207:2008, Information technology -- Software life cycle processes, Geneva: International Organization for Standardization (ISO), 2008.
- [8] ISO/IEC 15289:2006, Systems and Software Engineering -- Content of systems and software life cycle process information products (Documentation), Geneva: International Organization for Standardization (ISO), 2006.
- [9] ISO/IEC DTR 29110-5-1-2:2010, Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE) -- Part 5-1-2: Management and Engineering Guide - Basic Profile, Geneva: International Organization for Standardization (ISO), 2010.
- [10] ISO/IEC FDIS 29110-4-1:2010, Software engineering -- Lifecycle profiles for Very Small Enterprises (VSE) -- Part 4-1: Basic VSE profile specification, Geneva: International Organization for Standardization (ISO), 2010.
- [11] C. Laporte, Contributions to SE and the Development and Deployment of International SE Standards for Very Small Entities, PhD thesis, Université de Bretagne Occidentale, Brest, 2009, available at: <http://tel.archives-ouvertes.fr/tel-00483255/fr/> (last accessed May, 25th).
- [12] V. Basili, G. Caldiera, and G. Cantone, "A Reference Architecture for the Component Factory," ACM Transactions on SE and Methodology, vol. 1 (1), January 1992, pp. 53-80.
- [13] T. Chau and F. Maurer, "A Case Study of a Wiki-based Experience Repository at a Medium-sized Software Company," Proc. ACM K-CAP'05, ACM Press, 2005, pp. 185-186.
- [14] J. Rech, C. Bogner, and V. Haas, "Using Wikis to Tackle Reuse in Software Projects," IEEE Software, vol. 24 (6), pp. 99-104, Nov.-Dec. 2007.
- [15] C. Argyris and D. Schön, Theory in practice: Increasing professional effectiveness, San Fransisco: Jossey-Bass, 1974.
- [16] D. Schön, Educating the Reflective Practitioner: Toward a New Design for Teaching and Learning in the Professions, San Fransisco: Jossey-Bass, 1987.
- [17] E. Ras and J. Rech, "Using Wikis to support the Net Generation in improving knowledge acquisition in capstone projects," Journal of Systems and Software, vol. 82, April 2009, pp. 553-562.
- [18] N. Fenton, S.L. Pfleeger, and R.L.Glass, "Science and substance: A challenge to software engineers," IEEE Software, vol. 11 (4), July 1994, pp.86-95.
- [19] ISO/IEC DTR 29110-3:2010, Software Engineering - Lifecycle Profiles for Very Small Enterprises (VSE) -- Part 3: VSE Profiles Assessment Guide, Geneva: International Organization for Standardization (ISO), 2010.
- [20] ISO/IEC 15504:2004, Information technology -- Process assessment, Geneva: International Organization for Standardization (ISO), 2004.
- [21] S. Alexandre and C. Laporte, "Software Requirement Analysis," http://profs.logti.etsmtl.ca/claporte/VSE/Publications/DP-Software%20Requirements%20Analysis-V1_2.doc, Montréal, 2010 (last accessed May, 25th).
- [22] C. Argyris and D. Schön, Organizational learning : A theory of action perspective, Reading: Addison Wesley, 1978
- [23] C. Argyris, R. Putnam, and D. McLain Smith, Action Science, Concepts, methods, and skills for research and intervention, San Francisco : Jossey-Bass, 1985.
- [24] I. Chan and C. Chao, "Knowledge management in small and medium-sized enterprises," Communications of the ACM, vol. 51 (4), April 2008, pp. 83-88.
- [25] J. M. García Alonso, J. J. Berrocal Olmeda, and J. M. Murillo Rodríguez, "Documentation Center – Simplifying the Documentation of Software Projects," Proc. Wiki4SE Workshop – 4th International Symposium on Wikis, Porto, 2008..
- [26] J. W. Maxwell, "Using Wiki as a Multi-Mode Publishing Platform," Proc. 25th annual ACM international conference on Design of communication, ACM, New York, 2001, pp. 196-200
- [27] A. Rauschmayer, "Next-Generation Wikis: What Users Expect; How RDF Helps," Third Semantic Wiki Workshop. at ESWC, Redaktion Sun SITE, Aachen, 2009, poster.
- [28] V. Ribaud, P. Saliou, R. V. O'Connor, and C. Y. Laporte "Software Engineering Support Activities for Very Small Entities," Proc. 17th International Conference on European Systems & Software Process Improvement and Innovation (EuroSPI 2010), Springer-Verlag, September 2010.
- [29] A. H. Gold, A. Malhotra, and A. H. Segars, "Knowledge Management: An Organizational Capabilities Perspective," Journal of Management of Information Systems, vol. 18 (1), May 2001, pp. 185-214.
- [30] P. Saliou and V. Ribaud, "Bootstrapping an empty repertoire of experience: The design case," Proc. 1st Workshop on Human Aspects of SE (OOPSLA 2009), ACM, October 2009.
- [31] L. Gómez Arenas, "Deployment Package – Software Testing," <http://profs.logti.etsmtl.ca/claporte/VSE/Publications/DP-Software Basic Profile Testing-CL00.doc>, Montréal, 2010 (last accessed May, 25th).
- [32] ISO 9000:2005, Quality management systems -- Fundamentals and vocabulary, Geneva: International Organization for Standardization (ISO), 2005.