

# Utilization of a Set of Software Engineering Roles for a Multinational Organization

Claude Y Laporte<sup>1</sup>, Mikel Doucet<sup>2</sup>, Pierre Bourque<sup>1</sup>, and Youssef Belkébir<sup>3</sup>

<sup>1</sup>Department of Software and IT Engineering  
École de technologie supérieure  
1100, rue Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3.  
{Claude.Y.Laporte, Pierre.Bourque}@etsmtl.ca

<sup>2</sup>Center of Competence in Software Engineering  
Bombardier Transportation  
1101, rue Parent, St-Bruno, Québec, Canada, J3V 6E6  
Mikel.Doucet@ca.transport.bombardier.com

<sup>3</sup>ARINSO Africa  
219, BD. Med Zerktouni  
Angle BD. Brahim Roudani  
CP 20100 El Maarif – Casablanca, Morocco  
Youssef.Belkebir@arinso.com

**Abstract.** In this paper, we present the application of a set of software engineering roles. Role definitions were developed using internationally recognized software engineering reference documents for a major railway development organization: Bombardier Transportation. The description of the Software Architect role is explained. This paper will also illustrate how the role set could be used for project-specific needs during a typical project planning and launch session.

**Keywords:** Software Engineering, Roles definition, Process, standards.

## 1 Introduction

As stated by Humphrey [1]: “Without clearly identified responsibilities, it could take some time for a team to understand everything that it must do, to decide who should do each task... That is not so much because the engineers don’t want to take responsibility but rather because they don’t know what all the actions are or they are not sure whether anyone else is already doing them. They may also be reluctant to take on tasks that the team or team leader might plan to give to someone else.” This why we have conducted a project to improve the software engineering role definitions within the software engineering process definition of a large multinational organization. The project was also conducted because of the many conflicting and sometimes absent role definitions across the many Bombardier Transportation<sup>1</sup> sites and projects. Moreover, many of the roles that had already been defined were defined only very briefly.

This paper presents the utilization of a set of roles in a project that includes software development. A detailed analysis and improvement initiative in regard to the role definitions within the Bombardier Engineering System Software Engineering (BES SWE) process definition has already been conducted [2, 3]. The BES SWE is the common software engineering process definition of Bombardier’s Transportation division, in which each role definition specifies the purpose of the role, identifies the core responsibilities assumed by the role, and the hard and soft skills needed to perform the role.

Created in 1974 to provide subway wagons for the Montreal Transit Authority, Bombardier Transportation grew rapidly through many acquisitions to become the leading manufacturer of rail material for moving people. The company had 16,000 employees before Bombardier Transportation acquired, in 2001, ADtranZ. It is also interesting to note that ADtranZ was also the result of an upcoming merger between employees from sections of ABB and

---

<sup>1</sup> [www.bombardier.com](http://www.bombardier.com)

Daimler Chrysler. The acquisition of ADtranZ came with 20,000 employees with an engineering presence in 25 countries. The company also had to face many challenges of modern multinational organizations:

- Multidisciplinary system development,
- Multiple integrator-supplier relationships,
- Multi-country development,
- Multicultural teams,
- Downsizing/merger/turnover,
- Offshoring.

As an example of a complex software-intensive system, the company is working on the development of the European Rail Traffic Management System / European Train Control System (ERTMS / ETCS). This system will allow trains to cross borders without the need to change locomotive or driver. It also makes it possible for every train to be supervised individually, and for every train to be run according to its particular characteristics<sup>2</sup>.

In order to facilitate the identification and deployment of technologies, the company established a number of corporate Centres of Competence in various engineering specialties. The Centre of Competence (CoC) in Software Engineering, located just outside Montreal (Canada), is where this project was coordinated. Some of the tasks of the Software CoC are:

- To reduce technical risks and quality deficiency costs;
- To support and monitor strategic initiatives;
- To assess, develop and deploy (e.g. training) software engineering technologies such as processes (BES SWE), methodologies and tools.

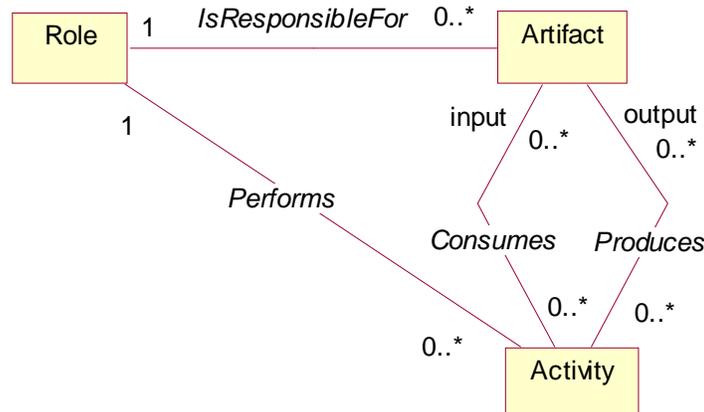
In order to provide technologies to all divisions, and at a rapid pace, it has been decided to use a common vocabulary, common processes and common roles. The strategy that was developed is as follows:

- Adopt internationally recognized reference documents
  - Models
  - Standards
  - Body of Knowledge
- Develop common processes, work instructions and role definitions
  - Independent of the organizational structure and organizational changes.

The notion of the role is a core concept in the BES SWE, as it is in all software engineering process definitions, as shown in Figure 1 (excerpted from OMG00). Roles perform activities that produce and consume artifacts. Roles are also responsible for artifacts. Of course, the same role may be performed in a given project by many people, and, conversely, one person may perform many roles. At the outset of the project, the BES SWE included 24 roles, as listed in Table 1, divided into four categories.

---

<sup>2</sup> Adapted from: *ERTMS/ETCS – for a competitive railway*, Bombardier Transportation, Rail Control Solution, Feb 2002.



**Fig. 1.** Interaction of roles, activities and artifacts in a software engineering process definition

**Table 1.** BES SWE role definitions at the beginning of the project

<p><b>Management Category</b>                      Senior Manager                      Project Manager                      Software Project Manager                      Software Quality Assurance                      Manager                      Product Manager                      Software Engineering Manager</p> <p><b>Software Engineering Category</b>                      Software Team Leader                      Software Requirements Coordinator                      Software Architect                      Software Implementer                      Software Integrator                      Software Test Designer                      Software Tester</p>	<p><b>Software Engineering –                      Supporting Category</b>                      Software Change Control Board                      Software Infrastructure                      Administrator                      Software Metrics Coordinator                      Software Process Engineer                      Software Project Coordinator                      Software Quality Assurance                      Engineer</p> <p><b>Other categories</b>                      Customer                      Proposal Coordinator                      Safety representative                      Software Trainer                      Software Training Coordinator</p>
---	--

In order to facilitate the roll-out of the role definitions to all Bombardier Transportation software engineering sites, it was decided that the coverage analysis and subsequent improvements to the role definitions would be founded on internationally recognized reference documents. Notably, this was viewed as a way of adding credibility to the improved role definitions without giving the impression that one software engineering site was imposing its role definitions on the other sites. The selected reference documents were IEEE/EIA Standard 12207.0-1996, Standard for Information Technology–Software Life Cycle Processes [4], the IBM Rational Unified Process (RUP) [5]<sup>3</sup> and the Guide to the Software Engineering Body of Knowledge (SWEBOK Guide) [6]. Each reference document is briefly described below.

### 1.1 IBM- Rational Unified Process

The IBM-Rational Unified Process (RUP) is a commercial object-oriented process framework for software development<sup>4</sup>. It includes a set of roles, activities, workflows and artifacts which describe the who, the how, the

<sup>3</sup> Version 2001A.04.00 of IBM RUP was used in this project.

<sup>4</sup> See <http://www-136.ibm.com/developerworks/rational/products/rup/>

when and the what of a software development process. It can be tailored to company specifics and to various sectors of the industry.

### 1.2 IEEE/EIA Standard 12207

The IEEE/EIA 12207.0-1996 Standard for Information Technology–Software Life Cycle Processes is considered a key standard in terms of the definition of life cycle processes. It has notably been designated as the pivotal standard around which the Software Engineering Standards Committee (SESC) is harmonizing its entire collection of standards. This standard groups software processes into activities and tasks, and these are organized into three categories: Primary Processes, which are divided into Acquisition, Supply, Development, Operation and Maintenance; Supporting Processes, which are divided into Documentation, Configuration Management, Quality Assurance, Verification, Validation, Joint Review, Audit and Problem Resolution; Organizational Life Cycle Processes, which are divided into Management, Infrastructure, Improvement and Training.

### 1.3 The SWEBOK Guide

The objectives of the SWEBOK Guide are to characterize the content of the software engineering discipline, to promote a consistent view of software engineering worldwide, to clarify the place, and set the boundary, of software engineering with respect to other disciplines, and to provide a foundation for curriculum development and individual licensing material. The SWEBOK Guide is a project of the IEEE Computer Society and has the support of numerous organizations<sup>5</sup>. The 2004 Version is also published as ISO Technical Report 19759 [7].

In the next section, an example of how one specific role definition was analyzed and improved based on the SWEBOK Guide is presented. The paper will also describe how the set of roles was used during project planning and launch activities in order to meet the specific needs of a particular project. A conclusion and ideas for further work are presented in the final section.

## 2 Comparing the Role Definitions and the Reference Documents: An Example

In the course of this project, every role definition was individually analyzed against each of the three reference documents. An example of such an analysis for the Software Architect role using the SWEBOK Guide as the reference document is found in Table 2. Such a table was completed for each role definition against each of the three reference documents. The detailed improvements on the role definitions are based on these tables. Table 2 is composed of the following cells:

- Role name: Name of the role in the BES SWE;
- OR: Overall recommendation on the presence of the role resulting from the analysis with the reference document (Accept, Remove);
- GAP: Overall evaluation of the difference between the definition of the role in the BES SWE and the definition of the role explicitly stated or implied in the reference document. Possible values are: Major, Minor, No Gap.
- RT: Recommendation regarding the role title (Accept, Modify);
- P: Recommendation regarding the role purpose section (Accept, Modify);
- CR: Recommendation regarding the core responsibilities section (Accept, Modify);
- HS: Recommendation regarding the hard skills section of the role definition (Accept, Modify);
- SS: Recommendation regarding the soft skills section of the role definition (Accept, Modify).
- BES SWE: Excerpted text from the definition of the role in the BES SWE prior to improvement.
- SWEBOK: Excerpted text from the SWEBOK Guide relevant to this role definition and potentially useful for improving the definition of the role in the BES SWE.
- Note: Indications on how to improve the definition of the role based on this comparison.

**Table 2.** Analysis of the Software Architect Role using the SWEBOK Guide as the reference document

<b>Role Name : Software Architect</b>	<b>Presence of the Role : Accept</b>
---------------------------------------	--------------------------------------

<sup>5</sup> Available free of charge on [www.swebok.org](http://www.swebok.org), and can also be purchased in book format from the IEEE Computer Society Press.

GAP : Minor	RT : Accept	P : Modify	CR : Accept	HS : Modify	SS : Modify
<b>BES SWE</b>		<b>SWEBOK</b>		<b>Note</b>	
<p>The Software Architect establishes the overall software architectural framework. Thus, in contrast with the other Roles (ex. Software Implementer), the Software Architect’s view is one of breadth, as opposed to depth.</p>		<p><b>Chapter 3 : Software Design</b>  <i>Software Structure and Architecture</i>                      In its strictest sense, “a software architecture is a description of the subsystems and components of a software system and the relationships between them”<sup>6</sup>.                      An architecture thus attempts to define the internal structure –“the way in which something is constructed or organized”<sup>7</sup> – of the resulting software.</p>		<p>The role of the Software Architect as and defined in the BES SWE assumes the activities stipulated in subsection III, Software Structure and Architecture, of the Guide SWEBOK</p> <p>The SWEBOK is very useful for improving the hard skills needed for this role.</p>	

The improved definition of the Software Architect role after the comparison with the three reference documents is found in Table 3 (proposed improvements are in *italics*).

**Table 3.** Definition of the Software Architect Role (proposed improvements are in *italics*)

<p><b>Purpose:</b></p> <p>The Software Architect establishes the overall software architecture. Thus, in contrast with the other Roles (e.g. Software Implementer), the Software Architect’s view is one of breadth, as opposed to one of depth.</p> <p><i>The Software Architect is responsible for articulating the architectural vision, conceptualizing and experimenting with alternative architectural approaches, creating models and components, interface specification documents, and validating the architecture against requirements and assumptions.</i></p> <p><i>Activities in this area include the creation of technology roadmaps, making assertions about technology directions and determining their consequences for the technical strategy, and hence architectural approach. This role involves not just these technical activities, but others that are more political and strategic.</i></p>
<p><b>Core Responsibilities:</b></p> <ul style="list-style-type: none"> <li>○ Derive the requirements for the system and software architecture.</li> <li>○ Identify the key design issues that must be resolved to support successful development of the software.</li> <li>○ Generate one or more alternatives and constraints for the architecture and select a solution.</li> <li>○ Allocate the software and derived requirements to the chosen architecture components and interfaces.</li> <li>○ Maintain requirement traceability for the software architecture’s requirements.</li> <li>○ Describe the software architecture by capturing the design results and rationale.</li> <li>○ Identify appropriate derived requirements that address the effectiveness and cost of life-cycle phases following development, such as production and operation.</li> </ul>

<sup>6</sup> Quotation from Chapter 6 of F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal. *Pattern-oriented Software Architecture – A System of Patterns*, J. Wiley and Sons.

<sup>7</sup> Quotation from the Oxford Dictionary.

**Hard Skills:**

- Ability to identify technical project risks based on the software architecture model.
- Ability to perform software modeling and architecture conception/definition.
- Ability to perform conceptual product design, *and specify a software architecture and implement a software system embodying it.*
- Ability to apply modeling techniques *such as use case, and other techniques, using the UML notations.*
- Ability to apply Architectural Styles, Reference models and Reference Architectures.
- Ability to specify structural descriptions with techniques and notations *such as: Architecture Description Language, Class Responsibility Card, Entity Relation Diagram, Interface Description Language, Jackson structure Diagrams and Structure Charts.*
- Ability to specify behavioral descriptions with techniques and notations *such as: Program Design Language PDL, Data Flow Diagram DFD and Flowcharts.*
- Ability to use computer-aided software engineering (CASE) tools in an architecture-driven design process.
- Knowledge in concepts of structural patterns *such as layers and client/server, mechanisms such as brokers and bridges, and middleware such as CORBA, DCOM, RMI.*
- Knowledge in operating system architectures, compiler and interpreter design, *and Real-time and Embedded Systems.*
- Ability to perform software engineering activities across the full development cycle, *including analysis, design, implementation, testing and documenting.*
- Understand the business context of Bombardier Transportation and its competitors, their products, strategies and product generation processes.

**Soft Skills:**

- Flexibility: The ability to adapt and deal with situations and manage expectations during periods of change.
- Sound Business Judgment: Know the business purpose of a project and make decisions within that context.
- Exhibit several communication styles: Be able to recognize a person’s communication style and adapt to it.
- Active listening skills.
- Setting and managing expectations.
- Conflict resolution.
- *Have the ability to make critical decisions under pressure.*

Table 4 and table 5 illustrate examples of how the consolidation and final decision of a few roles were formalized. For the software architect role, there were only minor gaps in both 12207 and SWEBOK. It was decided by Bombardier Transportation to keep the role and add the information gathered when performing the gap analysis.

**Table 4.** Examples of consolidation of analyses regarding the presence of the role<sup>8</sup>

Role name	RUP		IEEE 12207		SWEBOK	
	GAP	OR <sup>9</sup>	GAP	OR	GAP	OR
Project Manager	N	A	m	A	m	A
Safety Representative <sup>10</sup>	N/A	N/A	M	A	N/A	N/A
Software Architect	N	A	m	A	m	A
Software Engineering Manager	M	R	M	R	N	A

<sup>8</sup> Legend: M = Major ; m = minor ; A = Accept ; R = remove ; N = none ; N/A = Non Applicable

<sup>9</sup> Overall recommendation

<sup>10</sup> Safety functions are not covered in the IBM-RUP and are outside the scope of the SWEBOK Guide.

**Table 5.** Examples of consolidation of decisions regarding the presence of the role

Role name	Global recommendation of the study	Decision regarding the presence of the role by Bombardier Transportation	Rationale for the decision (when relevant)
Project Manager	A	A	
Safety Representative	A	A	Important role in the context of Bombardier
Software Architect	A	A	
Software Engineering Manager	A	A	This role is implied in the CMM [8].

### 3. Utilization of the Set of Roles during Project Planning and Launch Activities

Members of a new project usually have many concerns. In particular, they will have concerns regarding the organization of the team such as: Who will be my team members? Who will be the team leader? What will be my role and responsibilities? What will be the team members' roles and responsibilities? Will my team members have the skill and knowledge to do the project? Will we have all the skills to do the project?

To address these concerns, an organization may hold a project launch. A project launch is a workshop, usually led by a facilitator, in which identified project team members either define the project plan, including activities, deliverables and schedule, or walkthrough an already defined project plan. The project launch workshop could last between one and three days, but, for a typical Bombardier Transportation project, a one-day session is normally enough.

In order to illustrate the utilization of a defined set of roles, an example of a typical project planning and project launch session will be described. The purposes of a project launch session at Bombardier Transportation and are to:

- Define the project plan using an integrated team approach;
- Ensure a common understanding of objectives, process, deliverables, and role and responsibilities (R&R) of all project team members;
- Provide for an information exchange and offer just-in-time training to the project team members.

Before we present the example, an overview of the software engineering process will be provided to give the reader a better understanding of the environment at Bombardier Transportation.

#### 3.1 Overview of the BES SWE

The BES SWE was inspired by and partially derived from the Rational Unified Process (RUP)<sup>11</sup>. Illustrated in Figure 2, it provides a disciplined approach to assigning tasks and responsibilities within a software development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users within a predictable timeframe and budget. The BES SWE has two dimensions:

- The vertical axis represents processes, which are group of activities based on the IEEE 12207 standard. This dimension represents the static aspect of the process, that is, how it is described in terms of process items: processes/sub-processes, activities and artifacts.
- The horizontal axis represents time and shows the Life Cycle aspects of the process as it unfolds. This dimension represents the dynamic aspect of the process as it is unfolds, and it is expressed in terms of Phases, Iterations, Milestones and Formal Baselines.

To follow up on the set of roles described above, the Software Architect will be primarily responsible for the sub process titled Software Architecture Design. As illustrated in Figure 2, the Software Architect will have to

<sup>11</sup> Version 2001A.04.00 of RUP was used.

participate in formal reviews such as the Preliminary Design Review (PDR) and the Critical Design Review (CDR). This role will also either lead or participate in reviews such as Walkthrough and Inspection [9, 10].

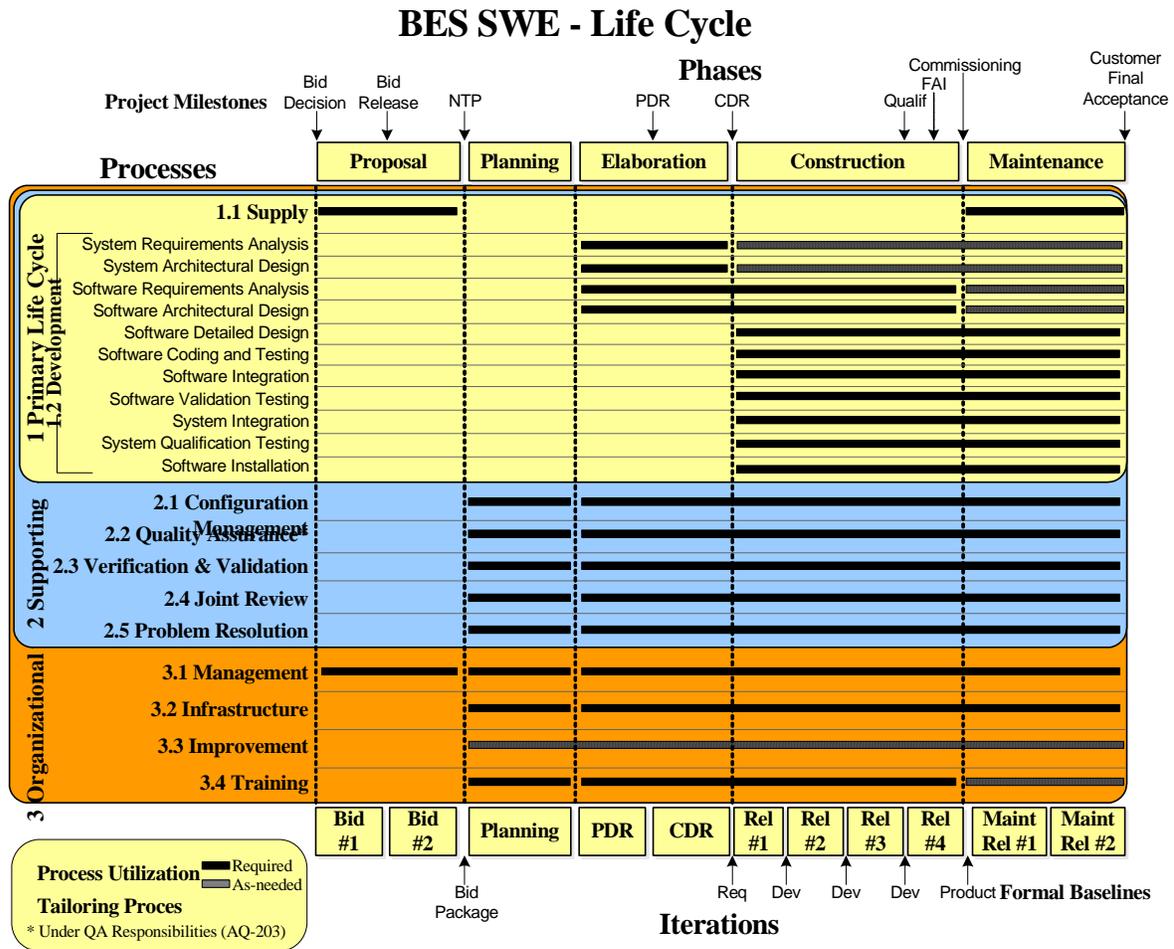


Fig. 2. BES SWE Life Cycle process<sup>12</sup>

### 3.2 Typical agenda of a project launch session

At Bombardier Transportation, a project launch session is typically performed at the beginning of a new project or at the beginning of an iteration. It can also be performed, for an iterative development project, to prepare the next iteration. In this case, it is called a project relaunch session. This intervention is also well suited in the case where a project's performance and/or process needs to be improved, when a project needs recovery, for example.

Depending on the size, complexity and type of project (e.g. new or modified/reuse, safety critical, etc.), a typical project launch session 'meeting' will last for 1 or 2 days at the same location. During a project launch session, it is important to have the team members' time 100% dedicated to this activity. In order to reduce office disturbance (e.g. phone calls), the project launch session may be held outside the project team office or building. Table 6 illustrates a typical timetable for a one-day project launch session. As shown in the table, under the topic Software Project Management (SPM) process, roles and responsibilities (R&R) are first discussed in item 4 and then in item 8. R&R

<sup>12</sup> Legend: NTP: Notice to Proceed  
 CDR: Critical Design Review  
 FAI: First Article Inspection  
 Dev: Development

PDR: Preliminary Design Review  
 Qualif: System Qualification  
 Req: Requirement  
 Maint: Maintenance

are also discussed under the topic Software Quality Assurance (SWQA) and Verification & Validation (SVV) processes.

**Table 6.** Typical activities and timetable of a project planning and launch session (1 day)

TIME	AGENDA ITEMS
08h30	Welcome, agenda review and participants' expectations <ul style="list-style-type: none"> <li>• Logistics: time-keeper and recorder roles</li> </ul>
09h00	BES Software Engineering Processes Overview
10h30	<b>Software Project Management (SPM) process:</b> <ol style="list-style-type: none"> <li>1. Identify/review all project inputs documents/information</li> <li>2. Project scope, constraints and assumptions</li> <li>3. Project iterations and associated objectives (imposed Milestones)</li> <li>4. Project team structure and Role allocation</li> <li>5. Project architecture (high level – SCI List)</li> <li>6. Project tailoring and deliverables list (per iteration) [WBS]</li> <li>7. Project staffing needs</li> <li>8. Intergroup relationships and associated Roles and Responsibilities (context diagram + R&amp;R table)</li> <li>9. Project risk identification and analysis</li> </ol>
12h00	Lunch
13h00	<b>Software Project Management (SPM) process (continue):</b> <ol style="list-style-type: none"> <li>1. Project tailoring and deliverables list (per iteration)</li> <li>2. Project risk identification and analysis</li> </ol>
14h30	Break
14h45	<b>Software Development (SD) process:</b> <ol style="list-style-type: none"> <li>1. Requirement definition: level, attributes</li> <li>2. Requirement traceability relationships</li> </ol>
15h00	<b>Software Configuration Management (SCM) process:</b> <ol style="list-style-type: none"> <li>1. Configuration Management (SCM) process:</li> <li>2. Project Software Configuration Identification (SCI)</li> <li>3. Project Baseline Plan (per iteration)</li> <li>4. Development of baseline approach (including tagging)</li> <li>5. SCM Audits, release management</li> </ol>
15h45	<b>Software Quality Assurance (SWQA) and Verification &amp; Validation (SVV) processes:</b> <ul style="list-style-type: none"> <li>• Identify SQA activities and associated R&amp;R</li> </ul>
16h00	<b>Software Infrastructure and Training:</b> <ol style="list-style-type: none"> <li>1. Project Development environment</li> <li>2. Project Validation/testing environment</li> <li>3. Project System Qualification environment</li> <li>4. Project training needs</li> </ol>
16h30	Session wrap-up
17h00	End

In many locations in Bombardier Transportation, R&R is informally allocated, and, most of the time, there is no name associated with the set of activities performed by an individual. In a few locations, some roles are partially defined. But, the name of the role and its responsibilities varies from one location to another. It was essential to be able to deploy a common software engineering processes to embed an R&R set in the process. When a project launch was conducted, it was then only necessary to mention that the R&R had been developed using internationally recognized frameworks. Team members were then ready to proceed without arguments about the names and responsibilities of the roles in the defined R&R set. Additionally, common definitions of roles were essential when people from different sites had to work on the same project. It was very easy for the project manager to prepare his staffing plan, as we will show below.

### 3.3 Project Tailoring

During the project planning and launch session, one item for discussion is ‘Tailor the Project’. The output from this tailoring activity will establish the project deliverables and the activities needed to develop the deliverables. Also, using the list of defined roles and responsibilities, roles for this project will be identified during the tailoring activity. As an example, if there are Safety regulations (e.g. CENELEC Standard EN50128<sup>13</sup>) imposed on a Project, ‘Software Verifier’, ‘Software Validator’ and ‘Software Safety Assessor’ roles will be identified. If, for the same project, software has to be acquired from a supplier, then a ‘Software Acquisition Coordinator’ role will also be needed.

### 3.4 Project organization

Once the Project Manager has identified which roles are needed for his project, this person will then select, out of the pool of resources available, the people for whom the hard and soft skills have been identified. The project manager will again use the set of role definitions to select the individuals who have the required characteristics. Without a defined set of role definitions, it would be harder for project managers to perform this activity.

The project manager, having identified the individuals, can now fill in the Project Organization Table, as illustrated in Table 7. If a few roles of a local development center have already been defined, then, at this stage, the local role names will be used for the local project plan.

**Table 7.** Subset of a Project Organization Table

Name of team member	Local Title	Relationship*	SW Project Manager	SW Task Leader	SW Req. Coordinator	SW Architect	SW Implementer	SW Integrator	SW Tester	SCCB	SW Infrastructure Admin	SW Process Engineer
Greg	...	C							?			
Per	Software Engineer	I		X	X	X	X	X	X			
Michael	Software Engineer	I					X					
Annie	Software Tester	I										
Guy	Manager, QA and Test	E										
Pierre	Tech	E										

Notes about this table:

- \* Relationship → C: Customer representative. E: External resource. I: Internal resource

This table contains only the list of Roles identified for a specific project after tailoring. The Project Manager has to verify that at least one person has been identified per Project Role. (i.e. each column needs to have at least one ‘X’). The Project Organization Table (Table 7) is also used for the completion of the Project Staffing plan. As illustrated in Table 8, the numbers in the cells represent the required head count per Role for a given period within a Project Iteration.

<sup>13</sup> CENELEC EN50128 - Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems

**Table 8.** Example of a subset of a Project Staffing Plan.

Roles	Iteration #1 Requirement		Iteration #2 Release 1		Iteration #3 Release 2	
	Months 1 -3	Months 4 -6	Months 7 -10	Months 11 -16	Months 17 -20	Months 21 -23
Software Project Manager	0.3	0.3	0.3	0.3	0.3	0.3
Software Task Leader	1	1	1	1	1	1
Software Architect	1	.8	.8	.5	.8	.5
Software Implementer – Database			1	1	1	1
Software Implementer –GUI			1	2	2	1

### 3.5 Training Plan

If individuals identified in the Project Staffing Plan do not have the required skills and knowledge, a training matrix is generated (see Table 9) using, again, the information obtained from the set of role definitions. The training matrix will be used to develop the training plan for the project. Once the topic of the course has been identified, the following information is entered in the table:

- In the “type” column, an indication as to whether the training session is a formal course, self-training, lectures, from an external organization, etc.
- In the “iteration” column, an indication as to when the training will take place.
- In the “duration” column, an indication as to the length, in hours or days, of the training session.

**Table 9.** Example of a subset of a Training Plan Matrix

Training Topics	Iteration	Duration [day]	Management	SW Project Manager	SW Task Leader	SW Req. Coordinator	SW Architect	SW Implementer	SW Integrator	SW Tester	SW Process Engineer	SW QA Engineer
UML Modeling	R2	2d				X	X		X	X		
Peer Review	R1	1d			X	X	X	X	X	X	X	X
Estimation	R1	2d	X	X	X							

Once the project is completed, a lessons-learned session is held to identify the project’s strengths and weaknesses. One of the issues analyzed is human resources (i.e. role allocation, skill sets, training). The output of the lessons learned exercise is used to update the set of role and responsibility definitions and also to modify the software engineering processes.

## 4. Conclusion

A project was conducted to improve the software engineering role definitions within the software engineering process definition of a large multinational organization. Detailed improvements were proposed to all role definitions, and an illustration of these improvements was presented for the Software Architect role.

It was also demonstrated how, at Bombardier Transportation, software projects tailor the set of roles to meet specific project needs. The utilization of the list of roles was illustrated for a project planning and launch session. It was demonstrated how roles and responsibilities were selected and allocated. Also, it was shown how the list was used to prepare a staffing plan and a training plan. The set of roles has been used in six project launch sessions. This has helped Bombardier Transport show to their customers that a well defined set of software roles is implemented in their projects. Customers are also very satisfied because the role set were developed using documents such as ISO and IEEE Standards.

The use of standards has also reduced significantly discussions during the deployment of processes; since employees and managers know the value and the credibility of standards.

To deploy a common software engineering process in many sites around the world, it is essential to embed a common set of roles and responsibilities in the process. When a project launch is conducted, it is much easier to deploy the set of roles by demonstrating that these roles have been developed using three internationally recognized frameworks. Individuals are then ready to proceed without arguing about the names of the roles or responsibilities assigned to, and the skills required by, each role. Additionally, the common set of roles is essential when people from different sites have to work on the same project. It is easier for the project manager to prepare his project plan and his training plan.

As more remote Bombardier Transportation sites are putting their effort together to develop and integrate software components, it is critical to have a common set of roles and responsibilities. Bombardier Transportation is among a minority of organizations that have documented these roles, as stated in a recent book [11]. The author indicated that fewer than 65% of the organizations that he assessed had documents describing roles and responsibilities.

As a result of the utilization of the role set, we have responded to the typical team members' concerns, such as: Who will be my team members? Who will be the team leader? What will be my role and responsibilities? What will be the team members' roles and responsibilities? Will my team members have the skill and knowledge to carry out the project? Will we have all the skills to carry out the project?

## References

- 1 Humphrey, W, Introduction to the Team Software Process, Addison Wesley, 2000. p 24
- 2 Belkebir, Y. Analyse et amélioration des définitions de rôles du processus d'ingénierie logicielle du centre de compétence en génie logiciel de Bombardier Transport, Department of Software and IT Engineering, École de technologie supérieure, Montréal, 2003
- 3 Laporte, C.Y., Bourque, P., Belkebir, Y., Doucet, M., Amélioration de la définition des rôles du processus de génie logiciel de la société Bombardier Transport, Revue Génie Logiciel, Number 72, March 2005, pp 43-52
- 4 IEEE/EIA 12207.0-1996 IEEE/EIA Standard Industry Implementation of International Standard ISO/IEC 12207: 1995 (ISO/IEC 12207) Standard for Information Technology Software Life Cycle Processes, Institute of Electrical and Electronics Engineers, 1998
- 5 Kruchten, P. The Rational Unified Process: An Introduction. Addison-Wesley, 2003
- 6 Abran, A., Moore, J.W., Bourque, P., and Dupuis, R. (eds.). Guide to the Software Engineering Body of Knowledge. IEEE Computer Society Press, 2004.
- 7 International Organization for Standardization. Software Engineering Body of Knowledge, Technical Report ISO/IEC TR 19759, 2005
- 8 Paulk, M., Curtis, B., Chrissis, M.B., & Weber, C.V. (1993). Capability maturity model for software version 1.1. Software Engineering Institute, CMU-SEI-93-TR-24
- 9 Gilb, T., & Graham, D. Software inspection. Addison-Wesley, Wokingham, U.K. 1993
- 10 IEEE 1028-2002 Software Reviews, Institute of Electrical and Electronics Engineers, 2002
- 11 Poulin, L., Reducing Risk with Software Process Improvement, Auerbach Publications, 2005