

Amélioration de la définition des rôles du processus de génie logiciel de la société Bombardier Transport

CLAUDE Y. LAPORTE, PIERRE BOURQUE, YOUSSEF BELKEBIR
& MIKEL DOUCET

Résumé : On présente ici l'analyse et l'amélioration des définitions de rôles et également l'identification des responsabilités appropriées, des connaissances, et des habiletés requises pour chaque rôle d'une équipe de développement de logiciels. Le concept de rôle renvoie à un ensemble de responsabilités définies qui pourraient être assumées par une personne ou par les membres de l'équipe de développement qui sont supposées intervenir dans le processus d'ingénierie logicielle. Durant un projet, les membres de l'équipe peuvent jouer différents rôles. L'analyse des définitions de rôles est basée sur la norme IEEE/EIA 12207.0-Technologie de l'information - Processus de cycle de vie logiciel, le processus unifié de la société IBM Rational Software, le processus d'ingénierie logicielle de la société Bombardier Transport et le Guide du corpus de connaissances en génie logiciel (SWEBOK), IEEE/IEA 12207.

Mots clés : processus de développement, rôles, amélioration, RUP, SWEBOK

1. INTRODUCTION

Cet article présente un travail, mandaté par le Centre de compétence en génie logiciel de Bombardier Transport, qui consiste à analyser et à améliorer la définition des rôles des différents acteurs qui interviennent dans le processus d'ingénierie logicielle de cette société. La notion de rôle, décrite selon le meta-modèle de Software Process Engineering Management (SPEM) [1], est considérée par Bombardier Transport comme un concept clé de son processus d'ingénierie logicielle. Il s'agissait de donner pour chaque rôle, les définitions ainsi que les connaissances techniques et les savoirs comportementaux requis pour exécuter adéquatement les tâches afférentes à chacun des rôles identifiés.

Cette analyse des rôles a pour objectif d'améliorer et de développer la définition des rôles formulée dans un artefact spécifique du processus d'ingénierie logicielle de Bombardier Transport. Ce document de définition des rôles

en ingénierie logicielle permettra d'uniformiser les façons de faire des 34 sites de Bombardier Transport situés au Canada, aux États-Unis, en Asie et en Europe.

Comme démarche de travail, le processus d'analyse des rôles s'est basé sur l'analyse des activités et les habiletés nécessaires au processus de génie logiciel. Ces activités sont stipulées dans les normes internationales et d'autres documents adoptés ou conçus par le Centre de compétence de Bombardier Transport. L'analyse des rôles identifiés consiste à faire une mise en correspondance entre le processus d'ingénierie logicielle de Bombardier Transport : BSEP (*Bombardier Software Engineering Process*), la norme IEEE/EIA12207.0®¹ [2], le Processus Unifié de IBM Rational RUP®² et le guide du SWEBOK³ (*Software Engineering Body of Knowledge*) [3]. Il s'agit de retracer et d'examiner chacun des rôles pour faire ressortir d'éventuels écarts quant à leur définition

pour améliorer l'actuel artefact de définition des rôles.

1.1 LA SOCIÉTÉ BOMBARDIER TRANSPORT

Bombardier Transport est un chef de file mondial dans la fabrication de matériel de transport sur rail. Sa gamme de produits comprend des véhicules de transports de passagers ainsi que des systèmes de transport sur rail. La société fabrique des locomotives, des wagons de marchandises et des systèmes de propulsion et de contrôle, et fournit des systèmes et des équipements de signalisation. Bombardier Aéronautique est un chef de file mondial dans la fabrication d'avions d'affaires et de transport régional. La société Bombardier, dont le siège social est situé à Montréal au Canada, compte un effectif de quelques 80 000 personnes dans 24 pays en Amérique, en Europe et en Asie Pacifique⁴.

1.2 LE CENTRE DES COMPÉTENCES EN GÉNIE LOGICIEL

Des Centres de compétence de Bombardier Transport ont été mis en place afin de réduire les risques techniques et les coûts associés aux manques de la qualité. Ils sont spécialisés dans les domaines d'ingénierie tels que l'aérodynamique, l'étude de bruit, la sécurité et le logiciel. Le Centre de compétence en génie logiciel a pour mission d'encadrer et d'offrir de l'expertise en génie logiciel à l'ensemble de l'organisation, en ce qui concerne les initiatives stratégiques. Il a comme mandat de faire des revues sur les biens artefacts des contrats critiques pour le compte des vice-présidences d'ingénierie, tout en proposant des actions visant la réduction des risques identifiés. Il a pour responsabilité de faire évoluer et améliorer les habiletés et les compétences techniques des ingénieurs. À cet égard il organise et offre des programmes de formation technique.

Étant donné le nombre de sociétés qui ont été acquises par Bombardier Transport, il devient important que les ingénieurs logiciels de chacun des centres de développement utilisent une terminologie commune, des rôles communs, des processus et des outils communs. Un rôle important du Centre de compétence en génie logiciel est d'uniformiser l'usage des processus d'ingénierie logicielle tout en restant conforme au processus d'ingénierie système de Bombardier Transport. C'est dans ce contexte qu'a été entrepris ce travail d'amélioration de la définition des rôles.

2. LE PROCESSUS BSEP

Le processus BSEP s'inspire d'un certain nombre de normes comme la norme IEEE/EIA12207.0 et le modèle CMM® (Capability Maturity Model) pour lequel le centre de développement logiciel de Saint-Bruno a été évalué au niveau trois de l'échelle de maturité⁵ CMM®⁶ [4].

Le développement de BSEP lui-même a suivi une approche structurée semblable à celle utilisée dans les projets de développement logiciel. Il s'agit d'analyser les composants existants réutilisables, d'identifier les produits logiciels COTS (*Components Off the Shelf*), de produire une architecture, de produire et valider l'information détaillée. Cette approche comporte les tâches suivantes :

- Production d'un inventaire de processus passés et existants, dans une perspective de réutilisation ;
- Identification des normes logicielles et des références primaires qui seront utilisées comme base pour le BSEP et qui favorisent l'utilisation des meilleures pratiques ;
- Identification de configuration qui sert d'architecture pour le BSEP. Le document de configuration inscrit tous les éléments du processus qui formeront le processus final. C'est à ce niveau qu'on fait l'association des types de rôles pour chaque processus défini selon la norme IEEE/EIA 12207 ;
- Élaboration pour chaque processus, d'un document Word qui décrit les éléments du processus ;
- Élaboration d'une version intranet, dès que les documents Word sont stabilisés et validés.

Le BSEP fournit une approche disciplinée pour assigner les tâches et les responsabilités dans l'organisation en ce qui a trait au processus d'ingénierie logicielle. Son but est d'assurer une production de logiciels de haute qualité qui répondent aux besoins de ses utilisateurs finals avec une planification et un budget prévisibles. La figure 1 illustre une représentation de cycle de vie BSEP⁷.

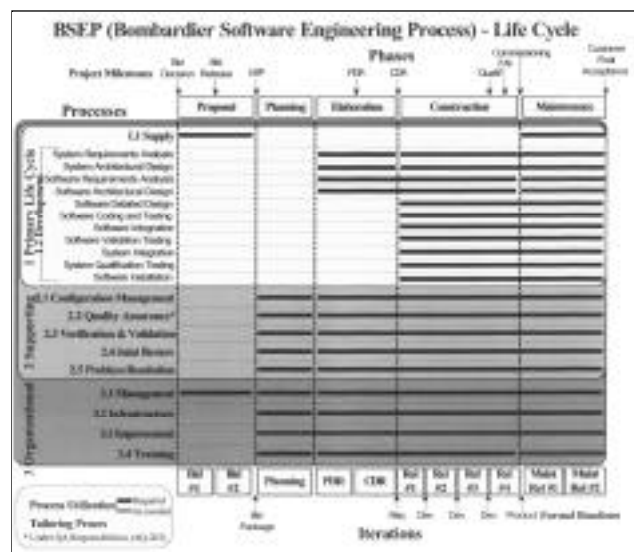


Figure 1. Cycle de vie du BSEP

Légende

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.

PROCESSUS

Inspiré de la norme IEEE/EIA 12207 et dérivé du processus RUP®, le BSEP a deux dimensions. La première dimension représente l'aspect dynamique du processus, elle est exprimée en termes de phases, d'itérations, de jalons et de bases de références. La deuxième dimension représente l'aspect statique du processus, elle est exprimée en termes de processus et d'activités stipulées dans la norme IEEE/EIA 12207.0. Les trois types d'éléments de processus de BSEP, décrits ci-dessous, sont les rôles, activités et artefacts.

2.1 LES RÔLES

Dans le contexte d'un projet d'ingénierie logicielle, un rôle définit les activités associées et les responsabilités d'un individu ou d'un ensemble d'individus travaillant en équipe. Le rôle avec les responsabilités associées définit les habiletés nécessaires pour effectuer les activités. Il faut considérer que les rôles n'identifient pas des individus précis ; néanmoins, ils décrivent les responsabilités d'un individu, et comment les individus devraient se comporter dans le projet. Comme métaphore du rôle, les membres individuels du projet peuvent jouer des rôles différents durant le projet ; c'est comme porter des chapeaux différents.

2.2 LES ACTIVITÉS

Les rôles réalisent des activités, qui définissent le travail qu'il faut effectuer. Une activité est une unité de travail qu'un individu avec la responsabilité décrite par le rôle sera invité à exécuter. L'activité a un but clair, d'habitude exprimé en termes de création ou de mise à jour de quelques artefacts. Les activités sont assignées à un ou plusieurs rôles. La granularité d'une activité est généralement de quelques heures à quelques jours et affecte un ou un nombre restreint d'artefacts. Une activité est utilisée comme un élément de planification et de suivi.

2.3 LES ARTEFACTS

Les activités ont en entrée et en sortie des artefacts. Un artefact est un produit de travail du processus : les rôles utilisent des artefacts pour réaliser des activités et pour en produire d'autres. Les artefacts peuvent être internes ou externes au projet.

2.4 LISTE DES RÔLES DÉFINIS DANS BSEP

Le tableau I énumère par catégorie, l'ensemble des rôles définis dans le processus d'ingénierie logicielle BSEP avant la réalisation du mandat décrit dans cet article.

2.5 GABARIT DES RÔLES

Dans l'artefact de définition des rôles, chaque rôle est détaillé à l'aide d'un gabarit qui précise le but, les principales responsabilités, les connaissances techniques et les savoirs comportementaux

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

Tableau I : Liste des rôles définis dans BSEP

Purpose : A Software Quality Assurance Manager fulfills the Role for all the Quality aspect for a project. The Software Quality Assurance Manager is the individual the Project Manager deals with in terms of Quality or Quality Assurance commitments and who controls all the Quality Assurance resources for a project.
Core Responsibilities : <ul style="list-style-type: none"> • Manage the SQA organization • Provide personnel to support the projects that require SQA activities • Do strategic planning for the SQA organization • Interview and hire SQA personnel • Inform senior management of the status of SQA and its activities across project supported • Monitor the SQA portion of proposals and estimates • Provide management interface with customer, engineering and production.
Hard Skills : <ul style="list-style-type: none"> • Ability to estimate, forecast, and plan the schedules, financial, physical, and human resources requirements • Understand the phases and components of the system life cycle • Understand the concepts covered in ISO 9000, Six sigma, Capability Maturity Model and process improvement.
Soft Skills : <ul style="list-style-type: none"> • 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 • Exhibits Several Communication Styles : Being able to recognize a person's communication style and adapt to it • Acts as a Coach and Mentor • Conflict Resolution

Tableau II : Exemple de définition de rôle

taux associés au rôle. Le tableau II montre un exemple de la définition, avant la réalisation de ce mandat, de rôle pour un gestionnaire d'assurance qualité logicielle.

3. DOCUMENTS DE RÉFÉRENCE UTILISÉS

Les trois documents utilisés, la norme IEEE/EIA 12207.0, le processus RUP® et le guide SWEBOK, sont reconnus internationalement comme crédibles, mais avec des objectifs complémentaires.

3.1 LA NORME IEEE/EIA 12207

Les normes de génie logiciel sont importantes dans la spécification des processus des entreprises. Généralement, elles sont employées afin de fournir un traitement systématique ou pour spécifier des techniques pour améliorer la production logicielle et accroître le niveau professionnel de la discipline d'ingénierie logicielle au sein de l'entreprise. L'Organisation internationale de normalisation (ISO) et la Commission électrotechnique internationale (CEI) ont publié la norme ISO/CEI 12207 Technologie de l'information - Processus de cycle de vie de logiciel. L'AFNOR a traduit et publié cette norme sous la mention NF ISO/CEI 12207 en 1995. La norme IEEE/EIA 12207.0, basée sur la norme ISO/CEI 12207, comprend des concepts et des

directives qui permettent une meilleure compréhension et application de la norme. Cette norme établit une structure commune pour des processus de cycle de vie logiciel, par l'application d'une terminologie bien définie. Elle définit les processus, les activités et les tâches qui devraient être appliqués pendant l'acquisition et le développement d'un système contenant du logiciel ou un produit logiciel autonome. Cette norme regroupe les processus dans trois catégories : de base, de support et organisationnel. Les processus de base sont les principaux processus initiateurs du cycle de vie logiciel. Ce sont l'acquisition, la fourniture, le développement, l'exploitation et la maintenance. Les processus de support sont : la documentation, la gestion de configuration, l'assurance de la qualité, la vérification, la validation, la revue conjointe, l'audit et la résolution de problème. Les processus organisationnels sont : le management, l'infrastructure, l'amélioration et la formation.

3.2 LES RÔLES DANS LE CYCLE DE VIE LOGICIEL SELON LA NORME 12207

L'utilisateur de la norme doit identifier les rôles joués dans un projet : acquéreur, fournisseur, développeur, opérateur, mainteneur ou gestionnaire. Du point de vue contractuel, l'acquéreur et le fournisseur négocient un contrat en utilisant respectivement le processus d'acquisition et le processus d'approvisionnement de fourniture. Du point de vue opérationnel, l'opérateur fournit le service d'opération de logiciel aux utilisateurs. Du point de vue ingénierie, le développeur ou le mainteneur effectue les tâches d'ingénierie pour produire ou modifier des produits logiciels. Du point de vue du support, les processus de documentation, de gestion de configuration, assurance qualité, de vérification, de revue conjointe, de validation, d'audit, et de résolution de problème sont employés pour fournir les services de support aux autres processus. Du point de vue organisationnel, ces processus sont employés au niveau de l'entreprise pour établir et mettre en œuvre une structure sous-jacente composée de processus de cycle de vie logiciel et les améliorations continues.

3.3 LE PROCESSUS UNIFIÉ DE LA SOCIÉTÉ IBM RATIONAL SOFTWARE

Le Processus Unifié de Rational (RUP)[®] est un processus d'ingénierie logicielle, qui fournit une approche disciplinée quant à l'assignation des tâches avec les responsabilités associées dans une organisation de développement logiciel. Le but du RUP, selon Kruchten [5], est de produire, dans les coûts et les délais prévus, des logiciels de grande qualité qui satisfont les exigences des utilisateurs finals.

Il fournit à chaque membre de l'équipe de développement des directives, des gabarits et des outils de mentorat ainsi que des conseils sur un grand nombre de techniques et d'approches modernes : la méthode objet, le développement à base de composants, la modélisation en UML, l'architecture et le développement itératif. Selon Kruchten [5], ces approches appelées « *best practices* », sont précisément identifiées et observées ; elles sont utilisées couramment par les organisations de développement performantes. Ces six pratiques sont : le développement itératif, la gestion des exigences, l'architecture à base de composants, la modélisation visuelle, la vérification continue de la qualité et la gestion des changements.

La structure du RUP comporte deux dimensions :

- L'axe horizontal, qui représente la dimension temporelle, montre le déroulement du cycle de vie du processus. Cette première dimension rend compte de l'aspect dynamique du processus qui s'exprime en termes de phases ;
- L'axe vertical représente les principaux enchaînements d'activités qui regroupent les activités selon leur nature. Cette seconde dimension rend compte de l'aspect statique du processus qui s'exprime en termes de composants de processus, d'activités, d'enchaînements, d'artefacts et de rôles.

La figure 2 présente la structure logique du Rational Unified Process (RUP). Cette figure montre les variations de l'effort temporel durant le processus.

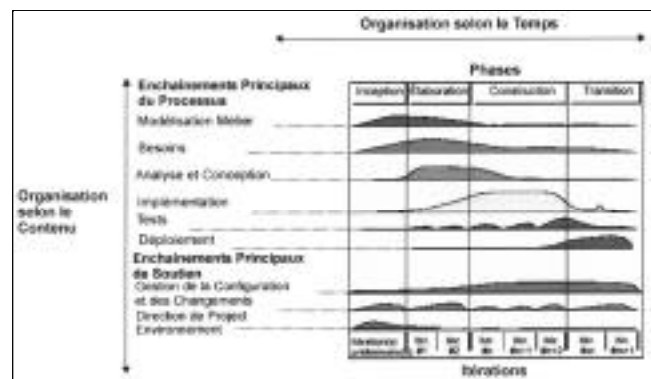


Figure 2. Structure logique du RUP

Les rôles définis dans le RUP

Le RUP introduit le concept du travailleur (*Worker*) qui définit le comportement et les responsabilités d'un individu ou d'un groupe d'individus travaillant dans une équipe. Le comportement s'exprime en termes d'activités que le travailleur accomplit. Le terme travailleur fait référence au rôle que doit tenir un individu dans le cadre de son travail et aux responsabilités afférentes, sachant qu'un individu remplit un ou plusieurs rôles et est responsable d'un ou de

plusieurs artefacts. Le concept de rôle est considéré comme « une casquette » dont un individu peut faire usage dans le projet. On s'attend à ce que l'individu désigné comme travailleur possède un ensemble de compétences pour passer d'un rôle à un autre.

Les rôles exécutent un ensemble d'activités cohérentes, ces activités sont étroitement rapprochées et fonctionnellement couplées. Chaque rôle est chargé d'exécuter un ensemble d'activités liées. Dans ce contexte, des activités seront dites liées lorsqu'il est préférable qu'un même rôle les réalise. En général, les responsabilités de chaque travailleur portent sur les artefacts qu'il crée, modifie ou contrôle. Le RUP organise les rôles en cinq ensembles que nous présentons dans le tableau III.

Sous-ensembles de rôle	Rôle
Analyste	<ul style="list-style-type: none"> ➤ System Analyst ➤ Business Designer ➤ Business-Model Reviewer ➤ Business-Process Analyst ➤ Requirements Reviewer ➤ Requirements Specifier ➤ Test Analyst ➤ User-Interface Designer
Développeur	<ul style="list-style-type: none"> ➤ Capsule Designer ➤ Code Reviewer ➤ Database Designer ➤ Implementer ➤ Integrator ➤ Software Architect ➤ Architecture Reviewer ➤ Design Reviewer ➤ Designer ➤ Test Designer
Testeur	<ul style="list-style-type: none"> ➤ Tester
Manager	<ul style="list-style-type: none"> ➤ Process Engineer ➤ Project Manager ➤ Change Control Manager ➤ Configuration Manager ➤ Deployment Manager ➤ Project Reviewer ➤ Test Manager
Autres rôles	<ul style="list-style-type: none"> ➤ Stakeholder ➤ Any Role ➤ Course Developer ➤ Graphic Artist ➤ Tool Specialist ➤ System Administrator ➤ Technical Writer

Tableau III : Les rôles définis dans le RUP

3.4 LE GUIDE SWEBOK

Le Guide SWEBOK a pour but de caractériser le contenu de la discipline du génie logiciel, de promouvoir à l'échelle mondiale une vue cohérente sur génie logiciel, de clarifier la place et de définir les frontières du génie logiciel vis-à-vis d'autres disciplines ainsi que d'offrir les fondements du cursus et de l'octroi de certificats de compétence professionnelle.

Ce guide est un projet de l'IEEE Computer Society et a le soutien de nombreux organismes. À la suite de la publication de la version d'essai en 2001 et d'une période de test, la version 2004 du Guide SWEBOK [3] a été publiée récemment. Il s'agit d'un document validé de façon consen-

suelle, accessible gratuitement à partir du site www.swebok.org ; il peut aussi être acheté sous la forme d'un ouvrage de librairie auprès de IEEE Computer Society Press. Cette version de 2004, qui a reçu l'approbation du Conseil d'Administration de IEEE Computer Society, sera publiée par l'ISO sous la référence ISO Technical Report 19759.

Le Guide SWEBOK vise différents publics répartis dans le monde entier. Tout d'abord, il est dirigé vers les organismes publics et privés nécessitant une vue cohérente du génie logiciel afin d'en définir l'enseignement et la formation [5], de catégoriser les emplois et d'établir des politiques d'évaluation et de plans de carrière [6]. Il s'adresse aussi aux besoins des ingénieurs praticiens des responsables de projets logiciels, et aux responsables des politiques publiques [7] en matière de règles relatives à la délivrance de certificats de compétence professionnelle [8]. De plus, les associations professionnelles qui définissent leurs propres règles, les enseignants qui établissent des politiques d'accréditation de cursus universitaires, les étudiants ainsi que les enseignants et instructeurs en génie logiciel sont engagés dans la définition de cursus [9, 10] et de contenus de cours [11] tireront profit de la consultation du Guide SWEBOK

Le Guide SWEBOK cherche à identifier et décrire le sous-ensemble des connaissances relatives au génie logiciel qui est généralement reconnu. Ces connaissances s'appliquent le plus souvent à la plupart des projets, et un large consensus en valide la valeur et la pertinence [12].

Le Guide SWEBOK est articulé autour de dix domaines de connaissances, dont les descriptions sont agencées de sorte à distinguer les différents concepts importants et permettre au lecteur de retrouver rapidement les sujets d'intérêt. Lorsqu'il retrouve ces dits sujets, le lecteur est renvoyé aux articles et chapitres d'ouvrages retenus en raison de leur présentation concise des connaissances. Les dix domaines, chacun faisant l'objet d'un chapitre du Guide SWEBOK, sont rappelés dans la figure 3.

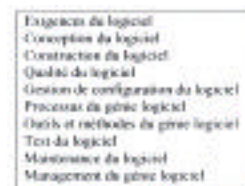


Figure 3. Domaines de connaissances du Guide SWEBOK

Pour rendre plus consistantes les définitions des rôles et plus spécifiquement les connaissances techniques du rôle, nous avons utilisé le

► Guide SWEBOK et le travail de Leonard L. Tripp [13] qui propose un modèle de compétence sur la définition des qualifications qui sont nécessaires en génie logiciel et qui sont basées sur les connaissances définies dans le Guide SWEBOK. Selon Tripp, le modèle de compétence consiste typiquement en : un ensemble de compétences nécessaires pour un rôle ou pour exécuter une tâche du travail, des qualifications requises pour réaliser chaque compétence et la connaissance requise à chaque compétence.

4. COMPARAISON ENTRE LES DÉFINITIONS ACTUELLES DES RÔLES ET LES DOCUMENTS DE RÉFÉRENCE

Les activités d'analyse et de comparaison sont effectuées en cinq étapes. Les cinq étapes sont :

1. Comparaison de la définition des rôles contenus dans le BSEP avec le RUP ;
2. Comparaison de la définition des rôles contenus dans le BSEP avec la norme IEEE/EIA 12207.0 ;
3. Comparaison de la définition des rôles contenus dans le BSEP avec le Guide SWEBOK ;
4. Révision de chacune des comparaisons par les experts ;
5. Mise à jour du document de définition des rôles de BSEP.

Le tableau IV explique chacun des champs du gabarit d'analyse.

Champ du Gabarit	Signification
Role Name	Nom du rôle dans BSEP.
Presence of the Role	Décision en fonction de l'analyse pour Accepter, Modifier ou Supprimer ce rôle.
BSEP	Définition du rôle proposé par le BSEP.
Mapping / Definition	Les rôles signalés dans le IEEE/EIA 12207, les définitions de RUP et les éléments du Guide SWEBOK qui traitent directement ou indirectement du rôle en question sont répertoriés dans la colonne (RUP, IEEE/EIA 12207, SWEBOK).
Note	Remarques substantielles à la définition de ce rôle.
GAP	Écart entre la définition du BSEP et celle identifiée dans le champ Mapping / Définition : Major, Mineur, Pas de Différence. Voir des liens demandés et des données pour les valeurs qui peuvent modifier ce champ.
RF	Rôle Title : Recommandation en fonction de l'analyse d'accepter ou de modifier le titre du rôle.
P	Purpose : Recommandation en fonction de l'analyse d'accepter ou de modifier.
CR	Core Responsibilities : Recommandation en fonction de l'analyse d'accepter ou de modifier les responsabilités associées au rôle.
HS	Hard Skills : Recommandation en fonction de l'analyse pour Accepter ou Modifier les connaissances techniques relatives au rôle.
SS	Soft Skills : Recommandation en fonction de l'analyse d'accepter ou de modifier les savoirs comportementaux afférents au rôle.

Tableau IV : Signification de chaque champ des gabarits d'analyse

Durant l'analyse, certains champs peuvent avoir les attributions N/A (n'est pas applicable) ou N/C (n'y a pas de changement à apporter).

4.1 DÉFINITION DU GAP

Nous ne disposons pas d'un formalisme rigoureux pour qualifier l'évaluation des écarts entre la définition ou les objectifs d'un rôle proposés par BSEP et celle provenant d'un autre élément d'analyse à savoir le RUP, la norme IEEE/EIA 12207.0 ou le Guide SWEBOK. Par conséquent, les comparaisons ont

été basées sur l'analyse de la pertinence et l'ampleur des activités et les objectifs associés à chaque rôle. Cette évaluation a une des trois mentions suivantes :

- *Majeure* : Une différence majeure signifie qu'il y a une différence notable quant à la portée des activités ou les objectifs associés à la notion du rôle ou bien une différence de nomination terminologique du rôle.
- *Mineure* : Une différence mineure signifie qu'il y a un certain degré de correspondance quant à la portée des activités ou les objectifs associés à la notion du rôle ou bien une différence mineure de nomination terminologique du rôle.
- *Pas* : Pas de différence : signifie qu'il y a un grand degré de correspondance et de complétude quant à la portée des activités ou les objectifs associés à la notion du rôle et la nomination terminologique du rôle.

Le tableau V montre un exemple du résultat de l'analyse du rôle *Software Architect* entre le document BSEP et la méthode RUP.

4.2 RÉVISION DES EXPERTS

Les trois tableaux d'analyse ainsi que le document de mise à jour final de la définition des rôles, ont été soumis à cinq révisions itératives. Au cours de chaque itération, il s'agissait de tenir compte des commentaires et remarques des révisseurs.

5. PRINCIPAUX RÉSULTATS

Un sommaire des principaux résultats de l'analyse portant sur l'amélioration et la validation de la définition des rôles est montré dans le tableau VI. Suite aux analyses effectuées antérieurement, il s'agissait d'attribuer, pour chacun des rôles, une des mentions suivantes : Accepter, Modifier ou Supprimer le rôle comme disposition finale de l'analyse. Pour chaque rôle et pour chaque type d'analyse, on a indiqué la mention concernant le *GAP* et la décision concernant la Présence du Rôle (*P. Role*).

Dans la colonne représentant la recommandation globale (*Global Recommendation*) nous avons attribué une recommandation, à savoir accepter, modifier ou supprimer le rôle. Cette recommandation a été attribuée en fonction des trois autres valeurs des colonnes *Présence du Rôle (P. Role)* : le rôle sera accepté s'il a au moins été accepté une seule fois tandis qu'une recommandation de suppression sera émise si le rôle n'a fait l'objet d'aucune acceptation. La colonne représentant la décision finale indique en définitive s'il faut accepter ou supprimer le rôle. Chaque décision est justifiée dans la colonne raison de la décision (*Rationale for decision*).

PROCESSUS

Role Name : Software Architect		Presence of the Role : Accept			
GAP : No	RT : N/C	P : Modify	CR : Modify	HS : Modify	SS : Modify
BSEP		RUP		Note	
The Software Architect establishes the overall software architectural. This, in contrast with the other Roles (i.e. Software Implementer), the Software Architect's view is one of breadth, as opposed to depth.		The software architect role leads and coordinates technical activities and artifacts throughout the project. The software architect establishes the overall structure for each architectural view; the decomposition of the view; the grouping of elements; and the interfaces between these major groupings. Therefore, in contrast to the other roles, the software architect's view is one of breadth as opposed to one of depth. The software architect must be well-rounded, possess maturity, vision, and a depth of experience that allows for grouping issues quickly and making educated, critical judgment in the absence of complete information.		This role process is similar objectives to those defined for the role of software architect defined in the RUP.	

Tableau V : Exemples d'analyse du rôle Software Architect.

- *Proposal Coordinator* : l'importance de ce rôle, provient du fait que le Coordinateur de proposition est responsable de réunir les membres de l'équipe pour la prise de décision d'offre (*Bid Decision go/no go*). Une décision d'offre, aboutit à la préparation d'une offre pour le projet potentiel qui amène à une soumission formelle auprès du client. De même, le rôle principal de coordonnateur de proposition est de préparer une proposition technique qui doit être approuvée par la direction.

Role name (in alphabetical order)	RUP		IEEE 12197		SWEBOK		Global consensus/endorsement of the study	Decision regarding the presence of the role by Bombardier Transportation	Rationale for decision (where relevant)
	GAP	UR	GAP	UR	GAP	UR			
Change Control Board	Minor	Accept	Minor	Accept	No	Accept	Accept	Accept	
Customer Product Manager	Major	Accept	Minor	Accept	No	Accept	Accept	Accept	The Product Manager usually works closely with Business Development.
Software Test Designer	No	Accept	Minor	Accept	Minor	Accept	Accept	Accept	
Software Tester	Minor	Accept	Minor	Accept	Minor	Accept	Accept	Accept	
Software Test Specialist	No	Accept	Minor	Accept	Minor	Accept	Accept	Accept	
Software Trainer	Minor	Accept	Minor	Accept	Major	Remove	Accept	Accept	This role is required in the field.
Software Training Coordinator	Minor	Accept	Minor	Accept	Major	Remove	Accept	Accept	This role is required in the field.

Tableau VI : Exemples de consolidation des analyses et des décisions.

- *Safety Representative* : l'importance de ce rôle émane aussi du fait que Bombardier Transport conçoit des véhicules de transport passagers ainsi que des systèmes de transport sur rail comme par exemple le contrôle des aiguillages. Par conséquent, le logiciel embarqué conçu est souvent de nature critique parce qu'il y a en jeu la vie des humains.

L'analyse a démontré que la quasi-totalité des rôles ont été acceptés ; seuls deux rôles ont fait l'objet d'une recommandation de suppression : Il s'agit du rôle de coordinateur de projet logiciel (*Software Project Coordinator*) et celui de n'importe quel rôle (*Any role*). Ces deux rôles présentent des GAP majeurs pour les trois types d'analyses.

Le tableau VII montre la définition du rôle *Software Requirements Coordinator*. Les améliorations apportées au rôle sont en italique.

La mention « modifier (*Modify*) le rôle » signifie apporter des améliorations dans la définition du rôle ; ces améliorations sont indiquées dans le rapport final et elles sont notées en caractères italiques afin de les distinguer du texte original.

6. TRAVAUX FUTURS

En ce qui concerne trois autres rôles (*Software Team Leader, Proposal Coordinator, Safety Representative*), nous avons évalué la pertinence de leur existence dans le processus d'ingénierie logicielle de Bombardier Transport. Les explications que nous avons données sur chacun de ces rôles se résument comme suit :

Le Centre de Compétence en génie logiciel de Bombardier Transport se prépare à déployer le processus BSEP de Bombardier Transport. De même, il est sur le point d'entamer des travaux d'investigations dans d'autres sites de Bombardier Transport situés en Europe, pour déterminer le profil individuel de compétences en génie logiciel. Le présent travail va aider beaucoup à évaluer le profil actuel pour spécifier le profil de compétence convoité.

Nous formulons deux propositions de travaux qui peuvent accompagner notre projet.

- *Software Team Leader* : le centre de développement logiciel de Bombardier Transport à Saint-Bruno (Québec) est déjà au niveau trois de maturité CMM, le rôle de leader d'équipe logicielle est une pratique courante voire une recommandation du CMM pour les organisations ayant un tel niveau de maturité logicielle.

Purpose: The Software Requirements Coordinator is responsible for requirements management of the overall software project. More specifically, the software requirements coordinator is responsible for eliciting the requirements and establishing and maintaining an agreement with the customer on the requirements of the software project. The software requirements analyst analyzes, elaborates and refines the allocated requirements to ensure that they are feasible and appropriate to implement in software, clearly stated, consistent with one another, testable, and complete.
Core Responsibilities: Responsible for the software requirements engineering process, requirements elicitation, requirements analysis, requirements prioritization, requirements validation, and requirements management. Responsible for requirements traceability and the generation of the Software Requirements Verification Traceability Matrix (SRV TM).
Hard Skills: <ul style="list-style-type: none"> • Ability to implement software requirements engineering processes; • Ability to acquire an understanding of the application and technology domain; • Ability to elicit software requirements from system stakeholders and to overcome common obstacles to the elicitation process; • Ability to describe mode- and operating condition requirements; • Ability to model software requirements using UML and CASE tools; • Ability to analyze and negotiate software requirements; • Ability to specify software requirements with relevant abstraction techniques; • Ability to perform architectural design and requirements allocation; • Ability to perform software requirements validation; • Ability to perform software requirements change management; • Ability to trace software requirements to software design artifacts; • Ability to trace software requirements to test artifacts.
Soft Skills: <ul style="list-style-type: none"> • Ability to negotiate and resolve problems when conflicts occur; • Active listening skills; • Flexibility: Ability to adapt and deal with situations and manage change; • Sound business judgment: Knowledge of the business process; • Exhibition of several communication styles: Ability to communicate with various stakeholders; • Setting and managing of expectations; • Ability to identify the key stakeholders.

Tableau VII : Définition du rôle Software Requirements Coordinator.

PROCESSUS

► Une première suggestion concerne l'amélioration du document de définition des rôles présenté à Bombardier Transport ; la seconde suggestion concerne le Guide SWEBOK.

Pour la première suggestion qui concerne BSEP, nos propositions sont :

- Améliorer BSEP en fonction du secteur de connaissances de la maintenance logicielle du Guide SWEBOK. Nous avons constaté à travers l'analyse faite qu'aucun rôle ne traite de la maintenance logicielle ;
- Inclure le rôle de rédacteur technique dans BSEP, comme le propose le RUP.

Comme deuxième suggestion, nous proposons la possibilité d'ajouter dans le Guide SWEBOK les deux points suivants :

- Offrir, en annexe du Guide, des définitions des rôles les plus appropriés et les plus pertinents, à chaque secteur de connaissances du Guide SWEBOK. Ce qui permettrait à d'autres organisations d'utiliser ces rôles à titre d'exemples.
- Inclure un chapitre ou traiter plus en profondeur la sûreté logicielle dans le Guide

SWEBOK, car même dans la norme IEEE/12207.0, nous avons trouvé très peu d'informations qui traitent du processus pour la sûreté logicielle.

Le tableau VIII présente une grille de couverture des rôles. La première colonne présente les différents rôles du BSEP, alors que la première ligne présente les dix secteurs de connaissances du Guide SWEBOK. Dans ce tableau, il s'agit de marquer les cases associées à chaque rôle et pour lesquelles il y a des activités qui sont couvertes dans un des domaines de connaissances de Guide SWEBOK.

Nous constatons, qu'effectivement il y a cinq rôles qui n'étaient pas entièrement traités, d'une façon ou d'une autre dans le Guide SWEBOK. Ce sont les suivants : le leader de l'équipe logicielle ; le coordonnateur de projet logiciel ; n'importe quel rôle ; le coordinateur de proposition et le représentant de la sûreté (*Software Team Leader ; Software Project Coordinator ; Any Role ; Proposal Coordinator ; Safety Representative*). Ces rôles sont marqués dans le tableau VIII en ligne grisée. Dans ce tableau, nous constatons

Domaine de connaissance du guide SWEBOK	Exigences	Conception	Construction	Test	Maintenance	Gestion de configuration	Management	Processus	Outils & méthodologie	Qualité
Senior Manager	√			√		√	√	√	√	
Product Manager	√			√		√	√		√	
Software Project Manager	√			√		√	√		√	
Software Quality Assurance Manager	√	√		√		√		√		√
Product Manager	√			√		√	√		√	
Software Engineering Manager	√			√		√	√	√	√	
Software Team Leader										
Software Requirement Coordinator	√			√					√	
Software Architect		√								
Software Implementer				√						
Software Integrator			√	√						
Software Test Designer				√						√
Software Tester				√						√
Software Change Control Board						√				
Software Infrastructure Administrator									√	
Software Metric Coordinator				√			√			
Software Process Engineer								√		
Software Project Coordinator										
Software Quality Assurance Engineer	√	√		√		√		√		√
Software Tool Specialist			√						√	
Any Role										
Customer	√									
Proposal Coordinator										
Safety Representative										
Software Trainer							√			
Software Training Coordinator							√			

Tableau VIII : Grille de couverture des rôles dans le Guide SWEBOK

aussi qu'aucun des rôles, parmi l'ensemble des rôles du BSEP, ne traite de la maintenance logicielle, car elle est considérée comme une version modifiée du cycle de développement. Dans la phase de maintenance, on peut compléter les fonctions non achevées ou reportées et aussi de corriger les défauts des artefacts.

7. RÉSUMÉS DES CONSTATS ET CONCLUSION

L'amélioration de la définition des rôles de BSEP a été effectuée à partir de trois documents internationalement reconnus : le RUP de la société IBM-Rational Software, la norme IEEE/EIA 12207.0 et le Guide SWEBOK. Bien que ces trois éléments concernent le génie logiciel, leurs objectifs sont largement différents. La norme IEEE/EIA 12207.0 décrit les activités du processus de cycle de vie logiciel aussi bien pour l'acquéreur que pour le fournisseur. Le RUP a été conçu pour être déployé par les praticiens du développement du logiciel, alors que le SWEBOK est un document de référence qui présente les connaissances généralement reconnues du domaine de génie logiciel. Pour atteindre cet objectif d'amélioration de la définition des rôles de BSEP, nous avons effectué notre analyse de sorte que chacun des éléments utilisés est orienté principalement vers une rubrique spécifique de la définition du rôle. Ainsi le RUP est orienté vers l'amélioration du but de la définition du rôle (*Purpose*), la norme IEEE/EIA 1207.0 est orientée vers l'amélioration des responsabilités associées au rôle alors que le Guide SWEBOK est orienté vers l'amélioration des connaissances et habilités associées à chaque rôle. De cette façon, nous avons pu améliorer l'ensemble des rôles de BSEP, aboutissant ainsi à la réalisation de l'objectif initial du projet.

Dans une certaine mesure, les résultats de cette étude d'amélioration de la définition des rôles de BSEP, au regard des trois éléments que sont la norme IEEE/EIA 12207.0, le RUP et le Guide SWEBOK, peuvent s'avérer difficiles à extrapoler à d'autres contextes, car l'élément central de l'analyse BSEP est un processus d'ingénierie logicielle propre à Bombardier Transport. Il présente la façon de faire adaptée au contexte d'affaires de Bombardier Transport. Par contre, les résultats de cette analyse, pourront être utilisés conjointement avec d'autres éventuelles définitions des rôles de façon à transporter ces définitions dans d'autres processus d'ingénierie logicielle similaires, parce que l'artefact de définition des rôles comporte des informations crédibles et provenant de sources solides.

8. RÉFÉRENCES

[1] *Software Process Engineering Management, the Unified Process Model* ; OMG

Document numéro ad/2000-05-05, mai 2000.

- [2] IEEE. 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.
- [3] A. Abran, J. W. Moore, P. Bourque et R. Dupuis (réds.) : *Guide to the software engineering body of knowledge* ; IEEE Computer Society Press, 2004.
- [4] M. C. Paulk et al. : *The Capability Maturity Model: Guidelines for improving the software process*. Addison-Wesley, Reading, Mass., 1995
- [5] D. Frailey et J. Mason : *Using SWEBOK for education programs in industry and academia* ; in 15th Conf. Software Engineering Education and Training Conference (CSEET 2002), (2002), 6-10.
- [6] S. McConnell : *Professional software development* ; Addison-Wesley, 2004.
- [7] T. Aytaç, S. İkiz et M. Aykol : *A SPICE-Oriented, SWEBOK Based, software process based assessment on a national scale: Turkish sector software survey - 2001* ; in 3rd International SPICE Conference, 2003.
- [8] OIQ. *Software Engineering Definitive Report* ; Ordre des ingénieurs du Québec, Montréal, Québec, 2000.
- [9] IEEE/ACM. *Computing Curricula - Software Engineering Volume - Public Draft 3.1, Joint Task Force on Computing Curricula - IEEE Computer Society/Association for Computing Machinery*, 2004.
- [10] S. Ramakrishnan et A. Cambrell : *An Informing Web-based Environment for a Bachelor of Software Engineering Degree - DoIT* ; in IS2002 Informing Science + IT Education Conference, 2002.
- [11] S. Ludi et J. Collofello : *An analysis of the Gap between the knowledge and skills learned in academic software engineering course projects and those required in real projects* ; in 31st ASEE/IEEE Frontiers in Education Conference, 2001.
<http://fie.engrng.pitt.edu/fie2001/papers/1187.pdf>
- [12] PMI : *A guide to the project management Body of knowledge* ; Project Management Institute, Newtown Square, Pennsylvanie, États-Unis, 2004.
- [13] Leonard L. Tripp : *Nominal software engineering competency model* (Discussion Version) ; mars 2002, Version 0.7

NOTES

- 1 Norme de la société IEEE.
- 2 Marque déposée de IBM Rational Software

- 3 Marque de service de la société IEEE.
- 4 Site Web de Bombardier Transport : www.Bombardier.com.
- 5 La date de l'évaluation remonte au 1^{er} février 2002.
- 6 CMM, Capability Maturity Model et Carnegie Mellon sont déposés auprès de l'U.S. Patent and Trademark Office par Carnegie Mellon University.
- 7 Document BSEP Global 000-SPD-0016.

BIOGRAPHIES :

- *Claude Yves Laporte* est actuellement professeur à l'École de Technologie Supérieure (ÉTS), une école d'ingénierie du réseau des établissements de l'Université du Québec, où il enseigne le génie logiciel. Avant de rejoindre l'ÉTS, il était (1992-1999) analyste senior auprès de la société Oerlikon Contraves, responsable de la coordination du développement et de la mise en œuvre des processus de management et d'ingénierie. Auparavant (1978-1991), il enseignait au Collège Militaire Royal du Canada, où il était aussi chargé, en tant que chef de projet, de conduire l'élaboration d'un programme d'enseignement de deuxième cycle en génie logiciel pour le compte du Ministère de la Défense Nationale du Canada. Ses centres d'intérêt portent sur le développement et le déploiement de processus de génie logiciel et d'ingénierie de systèmes, l'évaluation de processus, la qualité du logiciel et la conduite du changement. Il est membre de l'Ordre des Ingénieurs du Québec (OIQ), de l'International Council on Systems Engineering (INCOSE) et du Project Management Institute (PMI).
- *Pierre Bourque* est actuellement professeur à l'École de Technologie Supérieure (ÉTS), une école d'ingénierie du réseau des établissements de l'Université du Québec, où il enseigne le génie logiciel. Il fut, de 1995 à 2000, Directeur de la recherche appliquée au Laboratoire de recherche en gestion des logiciels de l'Université du Québec à Montréal. De 1987 à 1995, il était employé par la Banque Nationale du Canada où ses affectations ont porté sur l'évaluation des technologies de l'information, la conception de bases de données, le génie logiciel, la modélisation et l'administration de données. Il a publié et présenté des conférences sur la scène internationale portant sur le projet « Guide to the Software Engineering Body of Knowledge (SWEBOK), la mesure du logiciel, la mesure de la taille fonctionnelle du logiciel, les modèles de durée et les principes fondamentaux du génie logiciel. Il est

coéditeur du projet intitulé « Guide to the Software Engineering Body of Knowledge » de la IEEE Computer Society. Cette dernière lui a octroyé, en octobre 2001, un « Outstanding Contribution Award » pour son rôle dans ce projet. Il termine en ce moment un doctorat à l'Université d'Ulster en Irlande du Nord.

- *Youssef Belkebir* est actuellement consultant SAP-HR au sein de la société Arinso International. Ses centres d'intérêt portent sur la gestion des projets d'ingénierie du logiciel et sur la gestion des ressources humaines. Il détient un mastère de génie logiciel délivré par l'École de Technologie Supérieure à Montréal, un DEA électronique-traitement de l'information de l'Université des Sciences Hassan II au Maroc et un diplôme d'ingénieur en informatique de l'École Marocaine des Sciences de l'Ingénieur au Maroc.
- *Mikel Doucet* est responsable du Centre de Compétence en Génie logiciel chez Bombardier Transport, une entreprise d'envergure internationale et l'un des chefs de file mondiaux en solutions novatrices de matériel de transport sur rail. Le Centre de Compétence en Génie logiciel réunit près de 900 personnes réparties dans 30 groupes dans 12 pays et a comme mandat la définition et le déploiement de processus et d'outils communs, l'évaluation de risques et le support aux projets. Ce Centre de Compétence est aussi responsable de la gestion de réseaux d'échanges entre ces différents groupes. De 1995 à 1997, il était chef de la gestion de la configuration chez ATS Aérospatiale, responsable des activités relatives à la gestion de la configuration et aussi chargé du volet d'intégration et d'installation des systèmes chez les clients. De 1993 à 1995, il était ingénieur logiciel chez ADOPT Technologies, responsable de la conception d'interfaces graphiques pour des applications d'optimisation dans le domaine aérien. De 1991 à 1993, il a œuvré en tant qu'ingénieur en assurance qualité logiciel et, par la suite, ingénieur logiciel chez Systèmes Électroniques Paramax. Cette dernière société était le maître d'œuvre de l'intégration des systèmes électroniques de la Frégate de Patrouille Canadienne. Mikel Doucet a reçu une maîtrise en gestion de l'ingénierie de l'Université de Sherbrooke en 2002. En 1991, il a reçu un Baccalauréat en génie mécanique de l'Université de Sherbrooke. Il est membre de l'Ordre des Ingénieurs du Québec (OIQ), de l'International Council on Systems Engineering (INCOSE) et du Project Management Institute (PMI).