Coverage Analysis and Improvement of the Role Definitions of the Bombardier Software Engineering Process

Presented by Claude Y Laporte,
Professor - Department of Software Engineering and IT
École de technologie supérieure, Canada.

Agenda

- Introduction
- Portrait of Bombardier Transportation
- Challenges Facing Organisations
- Role Concept in the Bombardier Process
- Frameworks Used
- Methodology and Results
- Example of an Improved Role Definition
- Further Work

Bombardier Transportation

- A leader in the rail equipment, manufacturing and servicing industry.
- About 30,000 employees in 24 countries
  - Americas, Europe, Asia and Africa.
- Software Engineering Center of Competency (Québec) :
  - Established to reduce technical risks and quality deficiency costs.
  - Support and monitor strategic initiatives
  - Assess, develop and deploy (e.g. training) software engineering technologies.
    - e.g. Process (BES SWE), methodologies, tools.





Challenges

- Better, Faster, Cheaper
- Criticality of software
  - Financially, environmentally or for human safety.
- Multi-disciplinary system development,
- Integrator-Suppliers Relationships,
- Multi-country development,
- Multi-cultural teams,
- Downsizing/Merger/Turnover,
- Offshoring.

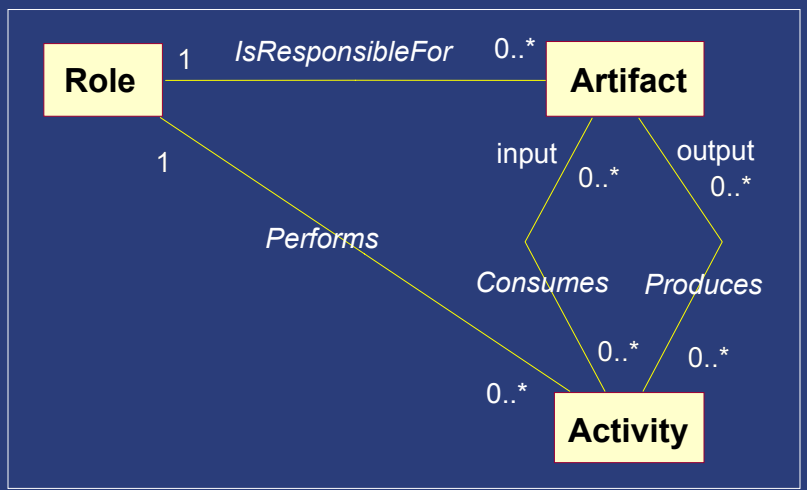
ERTMS / ETCS (European Rail Traffic Management System / European Train Control System)

Requirements and Strategy

- **Requirements**
  - Common Vocabulary
  - Common Processes
  - Common Roles
- **Strategy**
  - Adopt internationally recognized reference documents
    - Models
    - Standards
    - Body of Knowledge
  - Develop common processes, work instructions and role definitions
    - Independent from the organizational structure and organizational changes.

Role Concept

- Role defines the behaviour and responsibilities of an individual.
- Role is associated with:
  - Processes, Activities, Artifacts and Metrics.



Initial Role Definitions in BES SWE

- **Elements of Role Definitions**
  - Purpose
  - Core Responsibilities
  - Hard Skills
  - 'Soft' Skills
- **Roles Defined for Four Process Categories**
  - **Software Engineering**
    - e.g. Requirement Coordinator, Architect, Tester.
  - **Software Engineering Support**
    - e.g. Process Engineer, Quality Assurance,
  - **Management**
    - e.g. Software Project Manager,
  - **Others**
    - e.g. Trainer.

Implementation of Role Definitions in Software Process

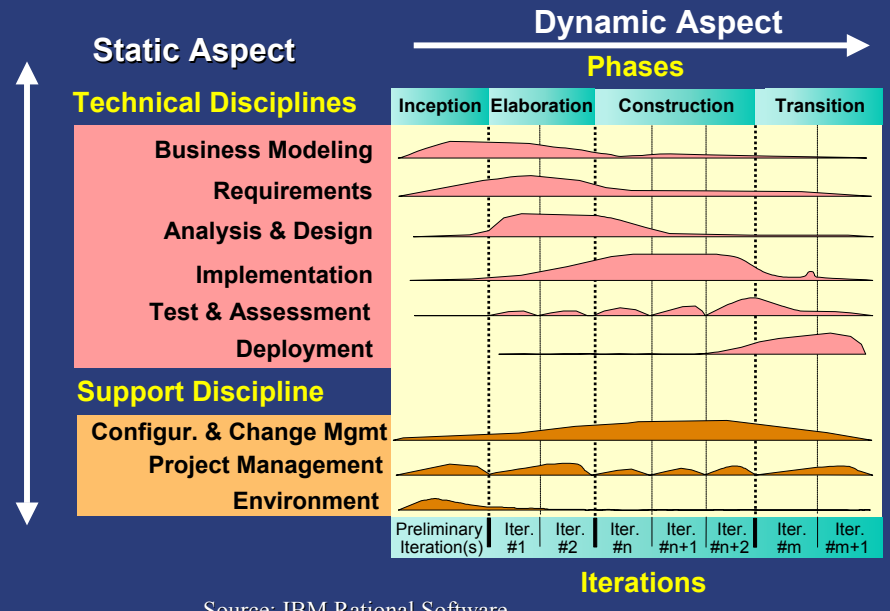
- Members of the organization may play different Roles
- Mapping from project individuals to Roles
  - Done during the initial project planning activities
  - Documented in the project plan

Source: IBM - RUP

Strategy to Improve Role Definitions

1. **Used internationally recognized software engineering reference documents**
   - IBM-Rational Unified Process (RUP),
   - ISO-IEEE/EIA Standard 12207,
   - ISO–IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK Guide).
2. **Mapped actual roles to each reference document.**
3. **Performed gap analysis**
   - e.g. Major, Minor, No Gap.
   - Provided rationale for decision
4. **Provided recommendations to Bombardier SWE CoC.**

IBM - Rational Unified Process (RUP)



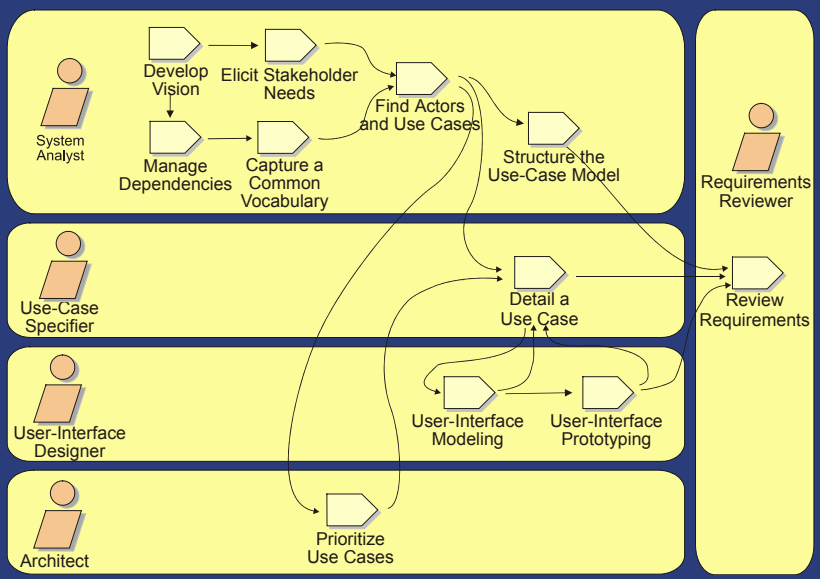
Source: IBM Rational Software

Roles in IBM-RUP

- Roles define the behaviours and responsibilities of an individual.

Analyst	• System Analyst • Business Designer • Business-Model Reviewer • Business-Process Analyst • Requirements Reviewer • Requirements Specifier • Test Analyst • User-Interface Designer
Other roles	• Stakeholder • Any Role • Course Developer • Graphic Artist • Tool Specialist • System Administrator • Technical Writer
Manager	• Process Engineer • Project Manager • Change Control Manager • Configuration Manager • Deployment Manager • Project Reviewer • Test Manager
Developer	• Capsule Designer • Code Reviewer • Database Designer • Implementer • Integrator • Software Architect • Architecture Reviewer • Design Reviewer • Designer • Test Designer
Tester	• Tester

Roles in the Requirements Workflow



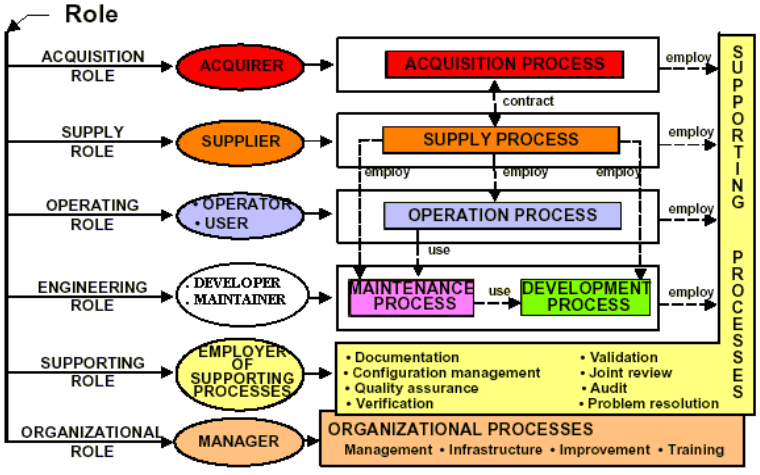
Source: IBM Rational Software

12207 Standard

- Framework for software life-cycle processes, with terminology that can be referenced by the software industry.
- An 'umbrella' standard
  - Standards are harmonized with 12207
- Defines processes, activities and tasks.
  - To acquire, supply, develop, operate, and maintain software products.
- Mainly used to provide activities and tasks in role definitions.

12207 Standard - Roles and Relationships

Processes and their relationships under key points of view



Source: Singh 95

SWEBOK

- **Sponsored by the IEEE Computer Society.**
  - Consensus on the core subset of knowledge characterizing the software engineering discipline.
  - Will be published as an ISO Technical Report (TR 19759).
- **Ten Knowledge Areas**
  - Requirements, Design, Construction, Testing, Maintenance, Configuration management, Engineering management, Engineering process, Engineering tools and methods, Quality.
- **Mainly used to improve hard skills needed for each role definition**.

Available free of charge at: www.swebok.org

Template for Role Comparison

Role Name :		**Overall Recommendation:**			
GAP :	**RT :**	**P :**	**CR :**	**HS :**	**SS :**
BES SWE		**RUP/12207/SWEBOK**		**Note**	
Abc…		**Abc…**		**Abc…**	

- **OR**: overall recommendation (Accept, Remove)
- **GAP**: (Major, Minor, No Gap)
- **RT**: role title (Accept, Modify)
- **P**: purpose (Accept, Modify)
- **CR**: core responsibilities (Accept, Modify)
- **HS**: hard skills (Accept, Modify)
- **SS**: soft skills (Accept, Modify)
- **BSEP**: Excerpted text from the definition of the role prior to improvement,
- **RPU/12207/SWEBOK**: Excerpted text potentially useful for improving the definition of the role.
- **Note**: how to improve the definition of the role

Actual Role Definition versus RUP

Role Name : Software Requirement Coordinator		Presence of the Role : Accept			
GAP : Mi	RT : Modify	P : Modify	CR : Modify	HS :Modify	SS : Modify
BES SWE		**RUP**		**Note**	
The Software Requirement Coordinator is responsible for the overall project Requirement management		The **system analyst** role leads and coordinates requirements elicitation and use-case modeling by outlining the system's functionality and delimiting the system; for example, establishing what actors and use cases exist, and how they interact. A person acting as system analyst is a good facilitator and has above-average communication skills. Knowledge of the business and technology domains is essential to have amongst those acting in this role. The **requirements specifier** role details the specification of a part of the system's functionality by describing the Requirements aspect of one or several use cases and other supporting software requirements. The requirements specifier may also be responsible for a use-case package, and maintains the integrity of that package. It is recommended that the requirements specifier responsible for a use-case package is also responsible for its contained use cases and actors		*This role is not defined in RUP.* This role is the aggregation in terms of activities of the two roles (System analyst and Requirements specifier Role) defined in RUP.	

BES SWE Versus SWEBOK

Role Name : Software Requirement Coordinator		Presence of the Role : Accept			
GAP : Mi	RT : Modify	P : Modify	CR : Modify	HS : Modify	SS : Modify
BES SWE		**SWEBOK**		**Note**	
The Software Requirement Coordinator is responsible for the overall project Requirement management		In this chapter the knowledge areas of software requirements is divided into six sub-areas : 1 Requirements engineering Process, it includes Process Models, Process Actors, Process Support and management, and Process Quality and Improvement. 2 Requirements Elicitations, it includes Requirement Sources, and Elicitation techniques 3 Requirements analysis it includes Requirements Classification, Conceptual Modeling, Architectural Design and Requirements Allocation, and Requirements Negotiation 4 Requirement Specification it includes Requirements Definition Document, Software Requirements Specification (SRS), Document Structure and Standards, and Document Quality 5 Requirements validation, it includes Conduct of requirements Reviews, Prototyping, Model validation, and Acceptance tests 6 Requirements Managements, it includes Changes Managements, Requirement Attributes, and Requirements Tracing.		The Guide SWEBOK uses the term of requirements engineer instead of software requirement coordinator. The Guide SWEBOK is very useful for improving the hard skills needed for this role.	

BES SWE Versus 12207 Standard

Role Name : Software Requirement Coordinator		Presence of the Role : Accept			
GAP : Mi	RT : Modify	P : N/C	CR : N/C	HS : N/C	SS : N/C
BES SWE		**IEEE/EIA 12207.0**		**Note**	
The Software Requirement Coordinator is responsible for the overall project Requirement management		**IEEE/EIA 12207.0 Clause 5.3 Development process : The Development Process contains the activities and tasks of the developer. The process contains the activities for requirements analysis, design, coding, integration, testing, and installation and acceptance related to software products. This process consists of the following activities :** **1) Process implementation;** **2) System requirements analysis;** **3) System architectural design;** **4) Software requirements analysis;** **5) Software architectural design;** **6) Software detailed design;** **7) Software coding and testing;** **8) Software integration;** **9) Software qualification testing;** **10) System integration;** **11) System qualification testing;** **12) Software installation** **13) Software acceptance support.** **5.3.4.1 « The developer shall establish and document software requirements, including the quality characteristics specifications, described below. Guidance for specifying quality characteristics may be found in ISO/IEC 9126…. ».**		The standard use the term of developer for those who perform the activities related to Software requirements as in the clause : 5.3.4.1 As mentioned in the clause 5.3, the role of developer is more generic. The overall activities of the development process are all in one role.

Software Requirements Coordinator *Modified* Role

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. He 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 specification, requirements validation, and requirements management.*

Responsible for requirements traceability and the generation of the Software Requirements Verification Traceability Matrix .

Software Requirements Coordinator *Modified* Role

Hard Skills:

- *Ability to implement software requirements engineering process;*
- *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 selected documentation techniques;*
- *Ability to perform software requirements validation;*
- *Ability to perform software requirements change management;*
- *Ability to trace software requirements to software design artefacts;*
- *Ability to trace software requirements to test artefacts*

Software Requirements Coordinator *Modified* Role

Soft Skills:

- *Ability to negotiate and resolve problems when conflicts occur;*
- *Active listening skills;*
- *Flexibility: Ability to adapt and deal with situations and manage expectations during periods of change;*
- *Sound business judgment: Knowledge of the business purpose of a project and decision-making within that context;*
- *Exhibition of several communication styles: Ability to recognize a person's communication style and adapt to it;*
- *Setting and managing of expectations;*
- *Ability to identify the key issues.*
- *Ability to acquire an understanding of the application and technology domain;*

Summary of Recommendations

GAP BES SWE versus	Major	Minor	No differences
IBM RUP	10	9	6
IEEE 12207	6	20	0
SWEBOK	11	11	3

- **Disposition of Recommendations by Bombardier**
  - **Approved all roles**
    - Improvement of all accepted role definitions,
  - **Two roles were removed, as recommended.**
    - The 'Software Project coordinator role' and 'Any role'
  - **A new role was approved, as recommended.**
    - Technical Writer.

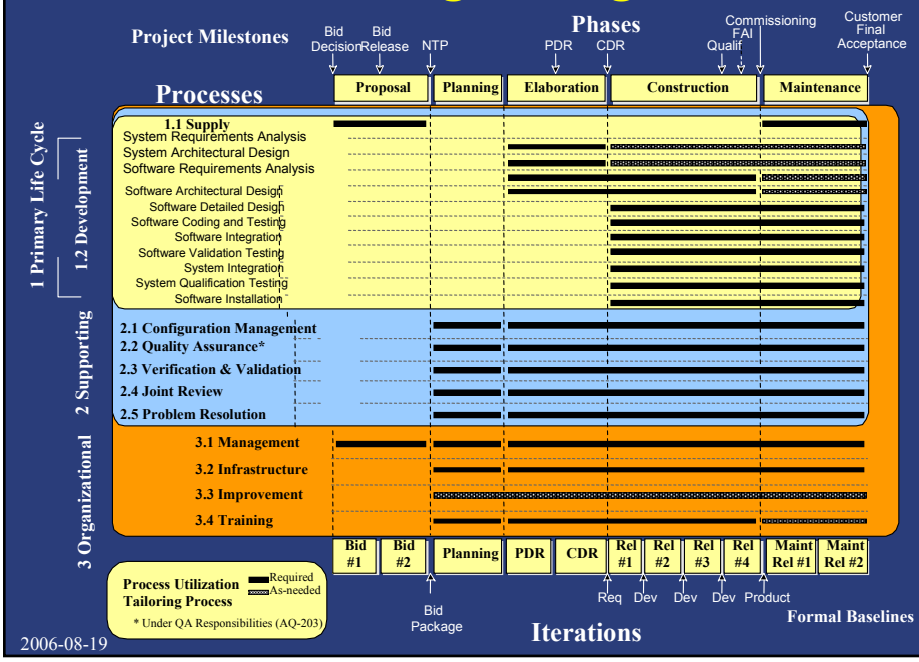
Further Work

1. Development of job descriptions for Human Resources,
   - e.g. hiring, promoting.
2. Development of training plan,
   - To fill skill and knowledge gaps,
   - To train new employees.
3. Improvement of Bombardier Process according to the SWEBOK Knowledge Area " Software Maintenance" ,
4. Development of a proposal to include role definitions in the SWEBOK,
5. Development of a proposal to include in the SWEBOK a new chapter about software safety.

Université du Québec
École de technologie supérieure

BOMBARDIER
TRANSPORTATION

Software Engineering Process



Bibliography

[1] Basili, V.R., R.W. Selby, D.H. Hutchens (1986). Experimentation in Software Engineering, IEEE Transactions on Software Engineering. Vol. 12, pp. 733-773.

[2] Bourque, P. (2000). Le cadre de Basili – Concepts, extensions et un exemple de son utilisation, MIG9250 – Séminaire de recherche [En ligne]. www.lrgl.uqam.ca.

[3] Abran, A., P. Bourque, J.-M. Desharnais, M. Maya, D. St-Pierre (1998). Designing of Functional Size Measurement for Real-time Software. Montréal, Université du Québec à Montréal, Montréal.

[4] Brouillette, M. (2000). Projet de maîtrise : Comparative analysis of the guide to the software engineering body of knowledge and the Rational Unified Process®. École de Technologie Supérieure.

[5] Duong, V. (2002). Projet de maîtrise : Un questionnaire d'identification des facteurs de risques des programmes d'amélioration du processus logiciel pour les organisations désirant progresser du niveau 1 au niveau 2 du modèle CMM. Université du Québec à Montréal.

[6] El Emam, Khaled (1995). Software Process Newsletter Committee on Software Process Technical Council on Software Engineering. IEEE Computer Society, TCSE no 3, Spring.

[7] Paulk, Mark, et al.(1994). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Reading, Mass.

[8] Software Process Engineering Management, the Unified Process Model OMG (2000). Document number ad/2000-05-05, May.

[9] IEEE/EIA 12207.0 1996 (1998). Standard for Information Technology-Software Life Cycle Processes. March.

[10] Singh, Raghu (1995). An Introduction to ISO/IEC 12207 (Tutorial). August.

[11] Singh, Raghu (1999). An introduction to International Standard ISO/IEC 12207 Software Life Cycle Process. FAA Washington DC, April.

[12] Moore, James W. (1998). Software Engineering Standards – A User's Road Map. IEEE Computer Society.

Bibliography

[13] A Rational Software Corporation White Paper (1998). *Rational Unified Process Best Practices for Software Development Teams*. TP-026A Rev. 11/98.

[14] Sten Jacobson Rational Software Scandinavia, *AB The Rational Objectory Process – A UML-based Software Engineering Process.*

[15] Jacobson, Ivar, Grady Booch, Jim Rumbaugh (1999). *Unified Software Development Process*. Addison-Wesley.

[16] Kruchten, Philippe (1999). *Rational Unified Process – An Introduction.* Addison-Wesley.

[17] Booch, Grady, Jim Rumbaugh, Ivar Jacobson (1999). *Unified Modeling Language—User's Guide*. Addison-Wesley.

[18] Royce, Walker (1998). *Software Project Management – A Unified Framework*. Addison-Wesley.

[19] Bourque, Pierre, Robert Dupuis, Alain Abran – Université du Québec à Montréal, James W. Moore – The Mitre Corporation & Leonard Tipp – The Boeing Company (1999). Guide to the Software Engineering Body of Knowledge. *IEEE Software*, Vol. 16, no 6, November/December.

[20] Bourque, Pierre, Robert Dupuis, Alain Abran, James W. Moore, Leonard L. Tripp (2001). *Guide to the Software Engineering Body of Knowledge trial version 1.0.* May [En ligne]. http://www.swebok.org/

[21] *A Guide to the Project Management Body of Knowledge* (PMBOK® Guide) (2000). Edition Project Management Institute, Newtown Square, Pennsylvania, USA.

[22] Tripp, Leonard L. (2002). *Nominal Software Engineering Competency Model (Discussion Version)*. March, Version 0.7.