Universitat Politècnica de Catalunya

Roles and Responsibilities for COTS Components Selection Processes Report Research

Fredy Navarrete, Pere Botella and Xavier Franch.
{fjnavarrete, botella, franch}@lsi.upc.edu
http://www.lsi.upc.edu/~gessi

Barcelona March 2006

Abstract

Information systems tend nowadays to be designed by integrating or customizing Commercial Off-The-Shelf (COTS) components acquired or licensed from the marketplace. There are currently many methods that contribute to select these components. The application of COTS selection methods result in processes that are different from usual development processes, yielding to new activities and responsibilities that should be covered by interactions of specialized roles. However it may be observed that these methods do not put emphasis neither on the identification of these roles, nor on their subsequent interactions, nor on their combination to form a selection team. The contribution of this work is based on identifying and defining the roles that take place in COTS selection processes, their interactions and their responsibilities. We use a goal-oriented approach, the *i** notation, and a framework to model the engineering process, the OPEN Process Framework (*OPF*), with the purpose of issuing a well-defined work team that can adapt itself to the internal processes of a particular organization. We apply our generic proposal to a particular case, a COTS-based development life-cycle based on the agile principles and individuals interactions as defined in eXtreme Programming (*XP*).

Contents

1. Introduction

Currently, organizations of every kind suffer from schedule and resource pressures and strong market competition when carrying out their daily business processes. Therefore, just a few of them have capacity and time enough to develop their own applications instead of acquiring or licensing Commercial Off-The-Shelf (COTS) components from the marketplace. Furthermore, there are also strategic or political concerns that drive organizations to opt for designing their information systems with COTS components. Consequently, the need to find the appropriate COTS component appears. Unfortunately, lack of structure and constant evolution has turned the COTS components marketplace in something nearer to a component bazaar than to a real specialized market [1]. Therefore, methods for the selection of COTS components (hereafter, *COTS selection methods*) have become necessary. As a result, different COTS selection methods have been proposed in the last ten years (CARE [2], SCARLET [3], OTSO [4], PECA [5], STACE [6], EPIC [7], among others). Although each of them has its own features and may vary in the proposed activities, all of them share the characteristic of differing radically from traditional software development methods and under some perspectives they still need to become mature.

In a previous analysis [8], we studied some COTS selection methods with the purpose of analyzing at what extent the agile principles and values briefed in the agile manifesto [9] were tackled. We observed that neither the human factors, nor the creation of a selection team, were clearly defined within the processes suggested by each of the studied methods. Also, we observed that COTS selection methods lack of a proper definition of the interactions that exist between systems stakeholders (e.g., the COTS component provider and the selection team). However, it has been reported that if we consider the human factor within the processes of an organization, we can obtain benefits in variables like time or cost [10]. Therefore, within COTS selection methods, we should incorporate concrete roles and responsibilities to be assigned to the members of a specialized work team. Those team members must have some abilities and specialized background [11], which emphasises the importance of the human factor within the development of the project.

The purpose of this paper is to identify and define the roles that take a part in COTS selection processes, stating their main responsibilities and their interactions. We base our work on two main information inputs. The first one is the review of some lessons learned reported in several COTS selection projects. The second input is the review of the roles that a few COTS selection methods have described explicitly. Once the roles are identified, we use a process-focussed OO methodology, the OPEN Process Framework (*OPF*) [12], to formalize them and their responsibilities. Next, we use a widespread goal-oriented notation, the i-Star (*i**) language [13], to put together the roles conforming a selection team, and to state the interactions among these roles, and also among the selection team and its environment, from a highly strategic perspective. Finally, we adapt this general framework to a particular case of COTS-based development life-cycle following the processes suggested by an agile method like eXtreme Programming (*XP*) [14], which emphasises the analysis of interactions among system stakeholders.

2. Lessons learned from cots projects

We have based our work on the concept of *anecdotal evidence* [15], which provides a context that allows interpreting the results reported by practitioners from their real projects. We have analysed under this perspective the lessons learned reported in several COTS projects [16 - 23] and as a result, we have identified some common needs that provide us a guide to suggest a set of roles and responsibilities that fit to these lessons learned. During this process, we have taken into account that anecdotal evidence depends on different factors that can impact either negatively or positively in the results of the report (e.g., whether in the selection project participated expert or novice personal).

In Table I we list the most relevant lessons reported in the COTS projects mentioned above that have to be with the human dimension of selection (in some cases, the impact on the human dimension of these lessons is direct, in others it is indirect), which may be summarized as:

- *To obtain flexibility in users requirements*: due to the constant evolution of the market and the little time available to select components, it is necessary a role steering the requirements capture, negotiating with the user the priority of his needs, being able to obtain flexible requirements that can be adapted to the current state of market. This role ought to maintain a fluid communication and good understanding with the stakeholders.

- *To involve the stakeholders in the selection team*: with the purpose of obtaining a constant validation and a good understanding of user requirements it is advisable that at least a representative of the stakeholders has a continuous participation in the selection team. Conversely, the stakeholder representative will be able to participate and to learn directly from the different processes results such as evaluation, selection and market exploration among others, which can help him to gain comprehension of the market offering.

- *To have techniques and software tools easy to use and learn:* inside the selection team it is important to include members with a verifiable technical level, able to use the set of techniques and software tools that may support selection processes. It is necessary that these techniques and tools are not rigid or difficult to implement, because their complexity can deteriorate the communication of the results making their use cumbersome.

- *To manage the relationships with the provider:* the work with the COTS provider company must seek the mutual benefit between it and the organization, avoiding to perceive the provider like a negative factor inside the project, because it is the one who provides a considerable percentage of component functionality [8]. For this reason, during the selection process a communication interface with the COTS provider is needed, helping the organization to communicate the needs to the provider.

- *To be sensitive to organization interests*: the contracts are the acquisition agreements of a market component, between an organization and a COTS provider. The set of clauses, duties and obligations in the contract must be aligned with the organization interests. For this reason, in the work team it is necessary a role with expertise in contracts.

TABLE I
LESSONS LEARNED IN SEVERAL COTS SELECTION PROJECTS

REPORT	LESSONS LEARNED
FAA SERC [16]	"…necessity of distinguishing between essential requirements and those that are negotiable"
	"Once a contract is signed, it can be extremely difficult to force a vendor to make good on their claims.
	"A common frustration was the lack of any real leverage to require vendors to live up to their claims."
L. Browns-word, P. Place [17]	"Operating in a commercial manner allows a greater use of COTS products."
	"Use contracts that provide flexibility for need identified after contract award."
M. Morisio et al. [18]	"Cleary, the number of deep evaluations must be kept low for cost and schedule reasons."
	"The team acts as a repository of history, knowledge and skills about COTS, and offers them to projects as a consulting activity."
C. Ncube, N. Maiden [19]	"Use the questionnaire in combination with other techniques to elicit initial supplier and product information"
	"There is often little time available for COTS software selection"
OSD [20]	"After a commercial item is selected, characteristics of the item and the vendor become integral parts of the system"
	"Commercial items can drive the system architecture and design"
D. Reifer et al. [21]	"Currently, few COTS software life cycle models address maintenance process for CBS"
N. Maiden et al. [22]	"Acquire in more detail those requirements that enable effective discrimination between products"
	"Stakeholder representatives should be present during product evaluation."
	"Better techniques are needed to record information during product evaluation."
	"Product evaluation is a team process, so treat it as such."
R. Adams, and S. Eslinger [23]	"Critical aspects of CBS development and sustainment are out of the control of the customer, developer and user."
	"CBS development and sustainment require a close, continuous and active partnership among the customer, developer and user."

3. Roles proposed by current cots methods

The existing COTS selection methods mentioned in the introduction [2 - 7] have their own set of practices and suggested processes. But as we have pointed out in a previous work [8], these methods do not fully succeed in considering individual motivations, as well as in defining the human factor, within their suggested processes. In fact, building a work team is considered a secondary aspect in conventional methodologies [15]. As a consequence, we cannot find in these approaches a selection team proposed formally. Nevertheless, a few of these approaches propose some roles explicitly [2, 5, 7, 24, 25] and we summarize them in table II.

TABLE II
DEFINITION OF ROLES IN SEVERAL COTS SELECTION METHODS

COTS Method	Roles suggested
CARE [2]	"Requirements engineer: complete and correct description of users needs with a technical background"
	"Software architect: defining the outline for the software architecture and defining the baseline software architecture"
	"Engineer component: maintaining the component repository"
	"Component vendor: complete and correct information about the component products"
IEEE 1062 Software Acqui-sition Stan-dard [24]	"Acquirer: a person or organization that acquires or procures a system or software product (which may be part of a system) from a supplier"
	"Developer: a person or organization that performs develop-ment activities (including requirements analysis, design, testing through acceptance) during the life cycle process"
	"Supplier: a person or organization that enters into a contract with the acquirer for the supply of a software product (which may be part of a system) under the terms of the contract"
EPIC [7, 25]	"Acquirer: identify relevant COTS packages and vendors"
	"System architect: oversees the entire system and all factors that might affect its development"
	"Designer: specify the COTS package configuration requirements"
	"Data designer: …mapping between the data sources and the target databases…"
PECA [5]	"Evaluator: should have technical experience"
	"Charter: defines the scope and constrains of the evaluation"
	"Evaluation stakeholders: are those individuals or groups with vested interest in the results of a COTS evaluation..."

4. A proposal of roles for COTS selection

Considering both the lessons learned and existing proposals of roles reported in the two previous sections, we present next our proposal of roles for forming a COTS selection team. We have identified nine roles specific of COTS selection:

A *System Architect:* Defines the structure of the information system, identifying constraints and technological specifications that compose it.

B *Market Watcher:* Explores the marketplace segments involved in the undertaken selection process to find the candidate COTS components which are to be evaluated and assessed with respect to user specifications.

C *COTS Component Evaluator:* Evaluates candidate COTS components which are assessed with respect to user requirements using the appropriate techniques. Experience in the component domain under evaluation is required.

D *Requirements Engineer:* Guides the elicitation, negotiation and validation of user requirements. To do so, he or she needs a minimum technical background and socialization ability.

E *COTS Vendor Interface:* Communicates with a particular COTS component provider company, trying to involve it inside the project, looking for mutual benefits of both parts.

F *Stakeholder Representative:* Someone who has an interest on the system-to-be and who, as a consequence, has an interest on the success of the selection process.

G *COTS Data Expert:* Evaluates and stores the information that is produced during the process, part of which may be used in future selection projects taking place in the same or similar domains.

H *COTS Lawyer:* Protects the company interests at the moment of acquiring or licensing a component, collaborating in the writing and review of acquisition contracts.

I *COTS Provider:* for providing detailed information and demos of components during detailed analysis;

TABLE III
MAPPING PROPOSED ROLES WITH EXISTING ONES

COTS Method	Roles Suggested	A	B	C	D	E	F	G	H	I
CARE [2]	Requirement engineer				**X**					
	Software architect	**X**								
	Engineer component							**X**		
	Component vendor					**X**			**X**	**X**
IEEE 1062 [24]	Acquirer						**X**			
	Developer	**X**		**X**	**X**					
	Supplier					**X**			**X**	
EPIC [7, 25]	Acquirer		**X**							
	System architect	**X**								
	Designer				**X**					
	Data designer							**X**		
PECA [5]	Evaluator			**X**						
	Charter			**X**						
	Evaluation stakeholders						**X**			

In Table III we show the correspondence of these proposed roles with the ones identified in Table II (each column stands for a role using the capital letters introduced above).

Besides this set of roles specific of COTS selection, other transversal ones can be incorporated in the selection team. Among them, we consider at least the Project Manager [7], the COTS Quality Engineer. The existence of these roles will be made explicit in the next section.

5. Roles formalization

In this section we aim at formalising the roles identified in the previous section. To do so, we need a notation or framework consolidated in the field of process modelling. OPEN is a framework created by a group of methodologists, researchers, tools vendors and practitioners [26], which includes concepts bound to business modelling, business decision making, maintenance, and application development. Our main purpose is to take the OPEN processes repository defined in [27] (OPF, the Open Process Framework) making stress in the responsibilities and tasks that should undertake the identified roles. In OPF, roles compose teams and these teams are part of organizations; we focus on the roles hierarchy without specifying what kind of organizations or what kind of teams the roles compose.

In figure 1, we present our COTS role hierarchy. We identify which roles are taken from the OPF (shaded boxes) and which are specific COTS roles (thick-lined boxes). These roles are classified according to two kinds of OPF roles, *Internal Role ("it is a producer internal")* and *External Role ("it is a producer external, outside of the work product to be developed but it is relevant to the development process")*. As a class of *External Role*, OPF proposes the *Representative* abstract class, which corresponds to a person that represents a specific type of organization or group of people that have common interests. Some of the roles identified in the previous section are defined as concrete classes that inherit directly from *Representative*:

- ***Vendor Representative*** (OPF), it is a representative of the COTS provider company (identified in a previous section as *COTS Provider*), with the purpose of providing detailed information and demos of components, among others benefices; ***User Representative*** (OPF), representing the needs of the stakeholders in the selection team (identified in a previous section as *Stakeholder Representative*); ***COTS Lawyer*** (COTS), this class inherits from *Partner Representative* (OPF) the knowledge about the contracts that are carried out with the *Vendor Representative* (OPF); ***COTS Vendor Interface*** (COTS), this class makes part of organization that requests a component, for this reason it inherits from *Customer Representative* (OPF); ***Market Watcher*** (COTS), this class inherits from two OPF classes. As a representative, it inherits from *Marketing Representative* (OPF)*,* because it uses the knowledge about the marketplace where the organization technology is developed. As an *Engineer* (OPF), it needs technical skills to perform marketplace exploration e.g. classifying technological segments from the marketplace.

On the other hand, among the *Internal Roles*, we find in OPF three abstracts classes of interest for our work. The first of them is *Engineer*, and as descendants we define:

- ***Requirements Engineer*** (OPF); ***COTS Quality Engineer*** (COTS), which helps in the definition of quality attributes of COTS components. This class inherits from *Quality Engineer* (OPF) the skills to provide improvement and time estimates of each selection activity; ***COTS Data Expert*** (COTS), this role inherits from *Database Engineer* (OPF), *Technical Writer* (OPF), *Database Architect* (OPF) and *Reuse Librarian* (OPF), because is the role for a person who creates, maintains and plans the data structure that will support the information that somebody can reuse inside the project or in future projects (for the project is very important storing the information that someone can use without documenting each process excessively); ***COTS Component Evaluator*** (COTS) and ***Market Watcher*** (COTS) are very-specific COTS roles that we define them as direct heir of *Engineer*.

The next abstract class that inherits from *Internal Role* is *Manager* (OPF), which makes reference to the administration activities carried out by a person. We find a class that inherits from *Manager*:

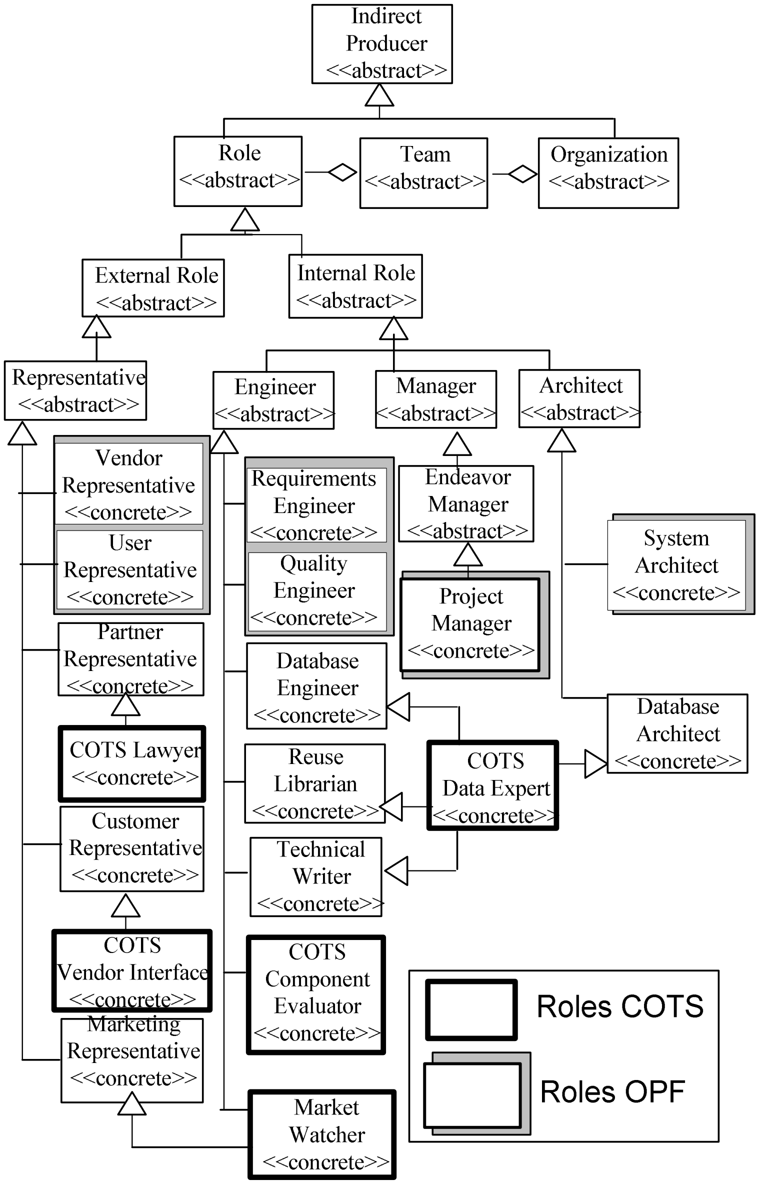
- ***Project Manager*** (OPF), which corresponds to the person in charge of representing the selection team at the organization. A person playing this role drives the work team through the selection process. This class inherits from other abstract class *Endeavor Manager* (OPF), because this class has the necessity of carrying out the project goals.

The final abstract class that inherits of *Internal Role* is *Architect*, which makes reference to the person that produces a specific architecture. We can find a concrete class in this hierarchy:

- ***System Architect*** (OPF), because this class has to describe the structure of information system.

As a result, we have defined all the roles identified in the previous section by contextualizing them in the OPF.

Figure 1. Formalization of roles using the OPF.


Indirect Producer <<abstract>>
Role <<abstract>>
Team <<abstract>>
Organization <<abstract>>
External Role <<abstract>>
Internal Role <<abstract>>
Representative <<abstract>>
Engineer <<abstract>>
Manager <<abstract>>
Architect <<abstract>>
Vendor Representative <<concrete>>
User Representative <<concrete>>
Partner Representative <<concrete>>
COTS Lawyer <<concrete>>
Customer Representative <<concrete>>
COTS Vendor Interface <<concrete>>
Marketing Representative <<concrete>>
Requirements Engineer <<concrete>>
Quality Engineer <<concrete>>
Database Engineer <<concrete>>
Reuse Librarian <<concrete>>
Technical Writer <<concrete>>
COTS Component Evaluator <<concrete>>
Market Watcher <<concrete>>
Endeavor Manager <<abstract>>
Project Manager <<concrete>>
COTS Data Expert <<concrete>>
System Architect <<concrete>>
Database Architect <<concrete>>
Roles COTS
Roles OPF

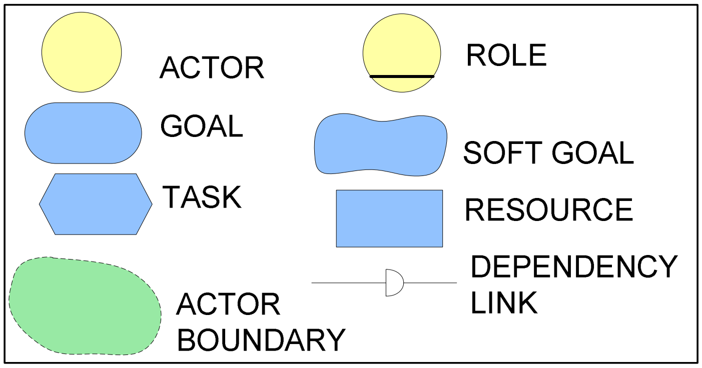
6. Roles Interaction

Once the roles that compose a COTS selection team have been clarified, we address to the different interactions that may occur among them. With this purpose, we use the *i** notation basically for two reasons: 1) it includes roles as part of its model elements; 2) it is possible to declare both high-level and low-level interactions, using the same model element (dependencies). For this reason, we use the Strategic Dependency (*SD*) models to identify the *Strategic Dependencies* that arise inside a selection team. We use the

R*i*SD methodology to construct this SD model [28], because R*i*SD suggests a construction guide and specific syntax for each constructor of an SD model.

The *i** consists of a set of nodes that represent roles and a set of *dependencies* that represent the relationships among them, expressing that an actor (*depender*) depends on some other (*dependee*) in order to achieve some objective (*dependum*). The *dependum* is an intentional element that can be a: *resource* (a physical or informational entity), *task* (particular way of doing something), *goal* (condition or state of affairs in the world that the actor would like to achieve) or *softgoal* (a condition in the world which the actor would like to achieve, but the criteria for the condition being achieved is not sharply defined a priori, and is subject to interpretation) [13, 28] (see Figure 2).

Figure 2. *i** notation.



In the fig. 3 we can observe the SD model that identifies the interactions among the members of a selection team. In the model we may distinguish the selection team (whose boundary is drawn in green) that contains the different roles defined in previous sections. Furthermore, some external actors appear, which represent the environment in which the selection team operates: the *Organization* interested in the selection, the *Information System* under construction, the COTS *Marketplace* and the *Vendor Representative* company. Dependencies among these actors and the roles inside the selection team are also included in the model.

We explain next, the most important interactions that appear in the model (we use the capital letters to identify the abbreviations of each role):

- *User Representative* (**UR**): depends on *Requirements Engineer* to validate his/her requirements, because the *Requirements Engineer* must negotiate and steer the user needs.

- *System Architect* (**SA**): depends on *Market Watcher* to compare the candidate components with the system architecture, for this reason the *Market Watcher* has to explore the marketplace to find components that will be evaluated.

- *Requirements Engineer* (**RE**): depends on *User Representative* to negotiate user requirements, because the *User Representative* has to adapt his/her requirements to the market.

- *Market Watcher* (**MW**): depends on *Requirements Engineer* to locate the candidate components, since the *Requirements Engineer* must define user requirements with the purpose of driving the search of components from the market.

- *COTS Component Evaluator* (**CE**): depends on *Market Watcher* to evaluate candidate components, because the *Market Watcher* must explore the marketplace to find components to be evaluated.

- *Vendor Representative* (**VR**): depends on *COTS Vendor Interface* to answer to the organization needs, since the *COTS Vendor Interface* is the communication bridge between the organization and the provider.

- *COTS Vendor Interface* (**VI**): depends on *Vendor Representative* to communicate him/her the project requirements, because the *Vendor Representative* can provide information and support to select a suitable component.

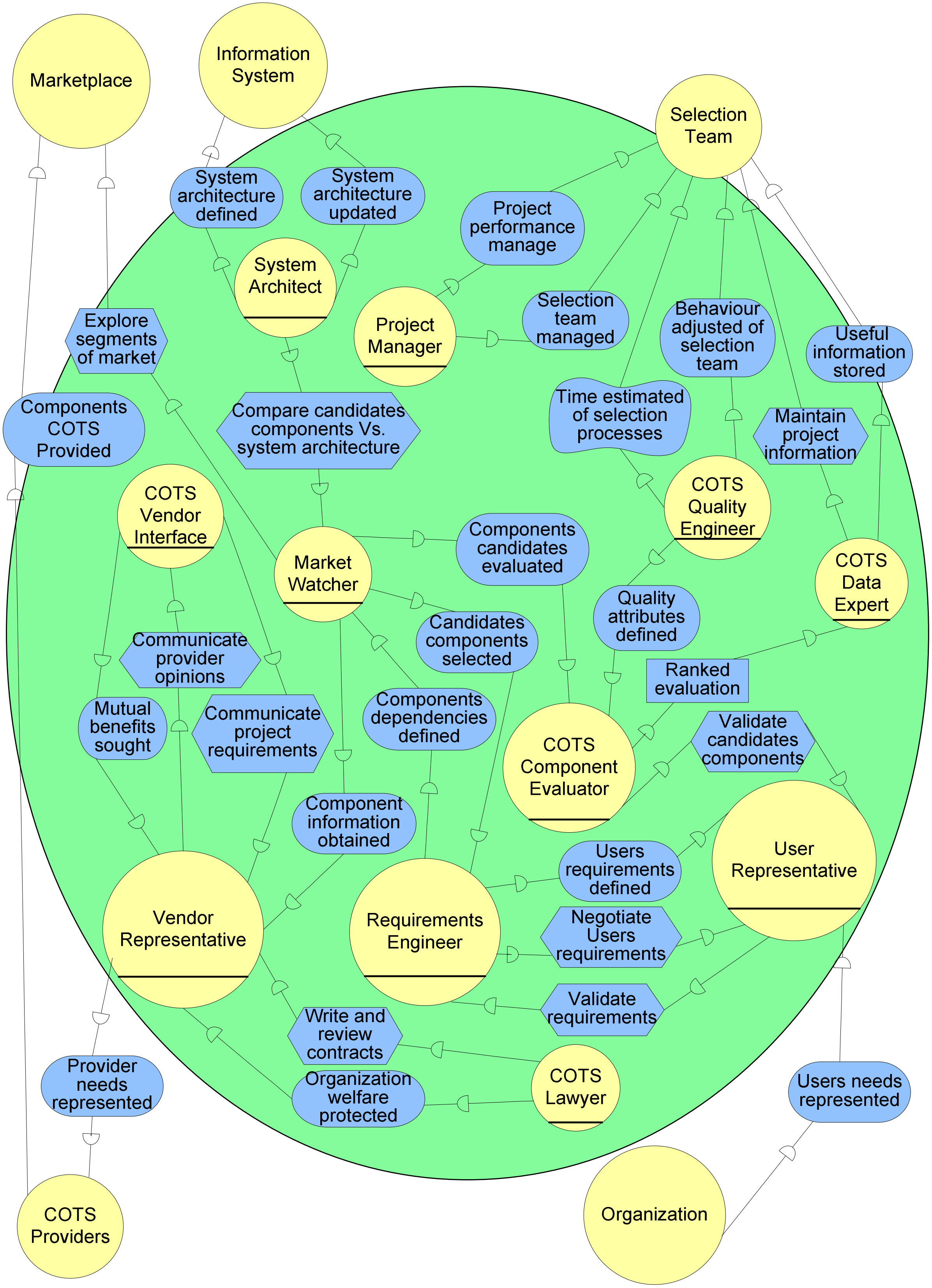
- *COTS Lawyer* (**LW**): depends on *Vendor Representative* to write and review the acquisition contracts since the *Vendor Representative* is the owner of components that the organization wants to acquire or license.

- *COTS Quality Engineer* (**QE**): depends on the whole selection team to estimate the time for each process, because the selection team has the knowledge for doing this task.

- *COTS Data Expert* (**DE**): depends on the whole selection team to store useful information, since the selection team must take advantage of this information.

- *Project Manager* (**MN**): depends on the whole selection team to manage it, because only with the help of the selection team the project manager can achieve his objectives.

Figure 3. i* model representing the roles and their interactions.


Marketplace
Information System
Selection Team
System architecture defined
System architecture updated
Project performance manage
System Architect
Project Manager
Selection team managed
Behaviour adjusted of selection team
Useful information stored
Explore segments of market
Components COTS Provided
Compare candidates components Vs. system architecture
Time estimated of selection processes
Maintain project information
COTS Vendor Interface
Market Watcher
Components candidates evaluated
COTS Quality Engineer
COTS Data Expert
Quality attributes defined
Communicate provider opinions
Candidates components selected
Ranked evaluation
Mutual benefits sought
Communicate project requirements
Components dependencies defined
COTS Component Evaluator
Validate candidates components
Component information obtained
Users requirements defined
User Representative
Vendor Representative
Requirements Engineer
Negotiate Users requirements
Validate requirements
Write and review contracts
COTS Lawyer
Organization welfare protected
Provider needs represented
Users needs represented
COTS Providers
Organization

7. An application: Agile COTS Selection

The aim of this section is to observe the different role interactions identified in the previous section in a particular setting that defines a concrete way of proceeding. For this reason we take as example a life-cycle method based on agile principles and values. In particular, we take an agile method that has good acceptance by the practitioners, XP [14]. XP is based on individual motivations and advocates the creation of a work team, which defines and identifies the necessary roles and responsibilities to carry out the project objectives. XP proposes 6 phases during its development cycle (*exploration, planning, iterations to release, productionizing, maintenance and death*). We can take these phases to extrapolate them inside a COTS context: *Exploration Phase*, *Planning Phase*, *Iterations Phase*, *Integration Phase*, *Maintenance Phase* and *Death Phase*. The scope of our work makes emphasis in the three first phases (*Exploration, Planning and Iteration* see Figure 4) which are the most related with COTS selection (we use the abbreviations introduced in section 6):

- *Exploration phase*: Identifies user needs, prioritizing and negotiating requirements with the support of techniques and technical tools for capturing the requirements. These tools must not be rigid nor complex, because it is necessary to have a good communication and a good understanding with the User Representative(by **RE, UR**). Also in this phase the high-level system architecture must define and highlight the system features and constraints to be compared with the user requirements (by **SA, RE**). At this point the exploration of the market must have started, defining the market segments where the organization technology is to be developed.

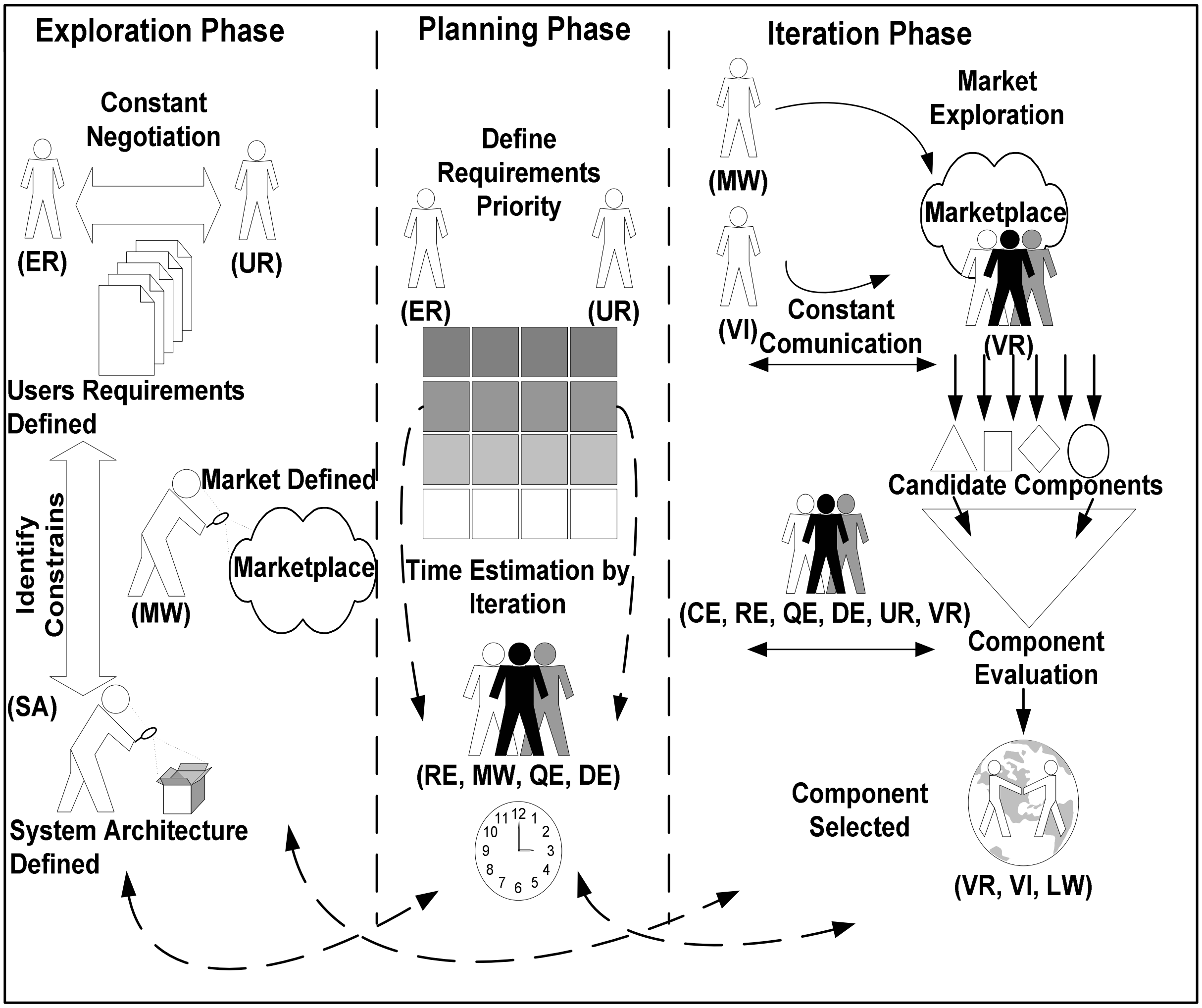
- *Planning phase*. The user highest priority requirements are defined (by **RE**, **UR**). Some estimates are made, for instance which is the time to find the **UR** requirements inside the market (by **QE, MW**). These estimates allow planning the appropriate number of iterations needed to search, evaluate and select COTS components. In this phase, knowledge enough to discern the marketplace segments of interest must be acquired (by **RE, UR, MW, DE**). In addition, dependencies from these identified segments to others (e.g., knowing that document management tools depend on document imaging systems to load hard copy documents into the system) shall be identified.

- *Iterations Phase*: The market exploration is started using the different sources of information available (by **MW, VI, DE**). When the market component candidates are identified, they must be evaluated (by **CE, QE, RE, UR**). If no COTS component satisfies the requirements up to an agreed extent, several options are possible: (a) make a best-of-breed solution in which different components are put together; (b) add an additional layer to a COTS component complementing its functionality; (c) develop a component entirely by hand; (d) adapt the STK requirements to the marketplace offering (by **RE, UR, MW**). In any case, if one or more COTS components satisfies the requirements these components must be evaluated (by **CE, RE, QE, DE, UR, VR**) to be selected, and then the legal and contractual aspects must be taken into account before considering the selection as definitive (by **LW, VI, VR**).

In these first 3 phases, the **MN** should steer the selection team, representing it and resolving internal problems. Also, the **DE** should evaluate what information will be able to be reused in each process to adjust the behaviour of selection team.

Adapting COTS selection processes to an existing paradigm may also make evident some synergies that result in new or improved practices. In the concrete agile case, and XP in particular, we may consider as new techniques and practices, among others: pair evaluation for obtaining at less two criterions about candidate components; system metaphor to define system features and restriction; component metaphor to identify suitable components; the planning game to plan the different iterations for evaluation; to adapt the story cards to capture of requirements.

Figure 4. COTS roles in agile COTS selection processes.



8. Conclusions

In selection processes of COTS components, it is necessary considering the creation of a specialized team for a suitable selection. Although different selection methods exist the interactions among their actors are not clearly defined. Therefore, using technical tools such as OPF and *i** to identify and define the roles involved in selection processes, their responsibilities and their interactions, provides an improvement in the maturity of COTS processes. On the other hand, if we consider adapting the COTS processes to development paradigms such as the agile one, we can obtain new possibilities to suggest new practices inside the COTS context.

Our research agenda is guided to define a maturity model for a whole COTS-based development life-cycle, where we can consider the integration and maintenance of the systems, also to propose a methodology that this based on the values and agile principles suggested in [9].

References

[1] B. Councill, S. Flynt, A. Mehta, R. Speed, and M. Shaw, "Component-based software engineering and the issue of trust." i*n Proc. 22nd International Conference on Software Engineering (ICSE),* Limerick, Ireland, 2000.

[2] L. Chung, K. Cooper, and S. Courtney, "COTS-Aware Requirements Engineering and Software Architecting" *in Proc. SERP* 2004.

[3] N. Maiden, H. Kim, and C. Ncube, "Rethinking Process Guidance for Selecting Software Components" *in Proc. 1st ICCBSS*, LNCS 2255, 2002.

[4] J. Kontio, "A Case Study in Applying a Systematic Method for COTS Selection" *in Proc 18th Intl' ICSE*, 1996.

[5] C. Dorda, C. Dean, E. Morris, and P. Oberndorf, "A Process for COTS Software Product Evaluation." *in Proc. 1st ICCBSS, LNCS 2255, 2002.*

[6] D. Kunda, "STACE: Social Technical Approach to COTS Software Evaluation" *in Proc. Component-Based Software Quality - Methods and Techniques*, LNCS 2693, 2003.

[7] C. Albert, and L. Brownsword, "*Evolutionary Process for Integrating COTS-Based System (EPIC): An Overview*". Carnegie Mellon University, Software Engineering Institute CMU/SEI-2002-TR-099 ESC-TR-2002-009, July 2002.

[8] F. Navarrete, P. Botella, and X. Franch, "How Agile COTS Selection Methods are (and can be)?" *in Proc. 31st IEEE EUROMICRO International Conference*, Porto, Portugal, 2005.

[9] Beck, K., *et al*. *Manifesto for Agile Software Development*. Available at: http://www.agilemanifesto.org

[10] K. Ohnemus, "*Incorporating Human Factors in the System Development Life Cycle: Marketing and Management Approaches*" in *Proc.* Professional Communication Conference (IPCC), 1996.

[11] C. Ncube, and N. Maiden, "PORE: Procurement Oriented Requirements Engineering Method for a Component-Based System Engineering Development Paradigm." in *Procs. 2nd International Workshop on Component-Based Software Engineering* (CBSE), 1999.

[12] D. Firesmith, B. Henderson-Sellers and I. Graham, *OPEN Modeling Language (OML) Reference Manual.* Cambridge Univ. Press, New York, 1998.

[13] E. Yu, "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" *in Proc 3rd IEEE Int. Symp. on Requirements Engineering, RE'97. Washington, USA*.

[14] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison Wesley, 1999.

[15] P. McBreen, *Questioning Extreme Programming*. Addison Wesley, 2003.

[16] FAA SERC. *Lessons Learned in Developing Commercial Off-The-Shelf (COTS) Intensive Software System October 2, 2000.* Available: http://www.faa.gov/aio/common/documents/HTMLfiles/LssLrnd.htm

[17] L. Brownsword, and P. Place, "Lessons Learned Applying Commercial Off-the-Shelf Products" CMU/SEI-99-TN-015, June 2000.

[18] M. Morisio, *et al.* "COTS-based software development: processes and open issues". Journal of Systems and Software 61 (2002): 189-199.

[19] C. Ncube, and N. Maiden, "Selecting COTS Anti-Virus Software for an International Bank: Some Lessons Learned" *in Proc. 1st MPEC Workshop, 2004.*

[20] Office of the Secretary of Defense (OSD). *Commercial Item Acquisition: Considerations and Lessons Learned On June 26, 2000.* Available: http://www.acq.osd.mil/dpap_archive/Docs/cotsreport.pdf

[21] D. Reifer, V. Basili, B. Boehm, and B. Clark, "COTS-Based Systems - Twelve Lessons Learned about Maintenance" *in Proc. ICCBSS 2004: 137-145*

[22] N. Maiden, C. Ncube, and A. Moore, "Lessons Learned During Requirements Acquisition for COTS Systems" Communication ACM 40(12): 21-25 1997

[23] R. Adams, and S. Eslinger, "Best Practices for the Acquisition of COTS-Based Systems: Lessons Learned from the Space System Domain" *in Proc. ICCBSS 2004: 203-205*

[24] *IEEE recommended practice for software acquisition*, IEEE Standard 1062, 1998.

[25] R. Pannone, and C. Peraire. The IBM Rational Unified Process for COTS-based projects: An introduction. 2005. Available at: http://www-128.ibm.com/developerworks/rational/library/aug05/peraire-pannone/

[26] B. Henderson-Sellers, "The OPEN framework for enhancing productivity" Software, IEEE 17(2), March-April 2000 Page(s):53 – 58

[27] D. Firesmith, B. Henderson-Sellers and I. Graham, *OPEN Process Framework (OPF).* Available: http://www.opfro.org/

[28] G. Grau, *et al*. "RiSD: A Methodology for Building *i** Strategic Dependency Models". In *Proc 17th International Conference on Software Engineering and Knowledge Engineering* (SEKE), 2005.