

# Reconciling Agility and Discipline in COTS Selection Processes

Fredy Navarrete, Pere Botella, Xavier Franch  
Universitat Politècnica de Catalunya  
{fjnavarrete, botella, franch}@lsi.upc.edu  
<http://www.lsi.upc.edu/~gessi>

## Abstract

*Currently, information systems are mainly built by integrating or customizing Commercial Off-The-Shelf (COTS) components acquired or licensed from the marketplace. The processes necessary to steer a suitable acquisition are different from traditional software development processes. Among them, we are interested in the process of selection of COTS components. COTS selection requires discipline to coordinate the selection team and the set of new activities that are necessary to support a successful selection. The Capability Maturity Model (CMM) contains important guidelines for process improvement, and specifies "what" we must have into account to achieve the disciplined processes (among others things). On the other hand, agile methods are playing nowadays an important role in software engineering practices, because they are specifying "how" the software practices must be addressed to attain agility and improvement in the software processes. The contribution of this work is to propose a framework to reconcile agile and discipline-based approaches in the COTS selection domain, by including agile practices into the 5 levels of CMMI for COTS acquisition.*

## 1. Introduction

The way in which information systems are being constructed has changed with the course of time. Organizations nowadays have neither time nor resources enough to develop their own applications. Furthermore, there are also some strategic or political concerns that drive these organizations to opt for designing their information systems with *Commercial Off-The-Shelf* (COTS) components. Developing COTS-based systems requires new processes to be undertaken and among them, COTS selection processes are critical.

The processes necessary to carry out a suitable COTS selection differ from those used during traditional software development. For this purpose, COTS selection projects need specific roles and new activities [1]. Currently, there are a lot of important lessons learned reported from several COTS projects which identify main concerns about selection processes, among others:

- There is a need for flexibility in defining requirements, because requirements engineering and COTS selection must be performed together [2 - 6].
- In selection processes, it is necessary to involve the system users and to work together with them to understand and comprehend their real needs [3 - 7].
- There is often little time available for COTS software selection, because it is required to operate in a commercial manner and a change in policy or in business processes may be requested at any time [3, 8].
- Understanding the marketplace is vital in COTS selection, because there is a need for continuous technology watch to keep up with vendors [2, 3, 9].
- Better techniques are needed for recording and managing information during COTS selection processes [5, 6, 7, 9].
- Often there is a lack of cooperation or trust with COTS vendors [2 - 5].

On the other hand, organizations nowadays have the opportunity to use some acquisition standards and COTS selection methods currently available to steer suitably the COTS selection processes. Nevertheless, these methods and standards are not used widely [10], because they propose technical tools and models that are not feasible or are unsuitable to be applied in industry context. Furthermore, in a previous work [11] we reported that COTS selection methods do not address important human aspects in their foundations which could improve basic features over COTS projects, such as collaboration, communication, and management.

Analysing the stated problems, it may be concluded that two general aspects that currently affect COTS selection processes are lack of maturity and lack of agility. The Capability Maturity Model (CMM) has wide acceptance in the industrial environment because of its improvement guidelines in the software process specifying “*what to do*” rather than “*how to do it*” [13]. On the other hand, the agile methods are generating interest in the industry by the importance of their software development practices, which refer to “*how*” we can drive the software processes to obtain agility. For this reason, the purpose of this paper is centred in suggesting a framework where we could obtain mutual benefits for using together maturity models and agile methods, taking advantage of the strengths of both contexts to apply them in COTS selection process specifying *what we can do* (with CMM) and *how we can do it* (with the best practices of agile methods) to obtain a successful COTS selection, and so providing coverage over main lessons learned reported from COTS projects.

The structure of this work begins with the background about the CMM and agile contexts focusing in the COTS domain. Then, we analyze the most widespread COTS selection methods to study their main processes and activities that could be influenced by the CMM and agile contexts. Next, we define the initial reconciliation point to work with both contexts in COTS selection domain. Afterwards, we propose the framework. Finally, we present the conclusions for our work.

## 2. Background

In this section we present some of the most relevant contributions performed over the COTS domain in the CMM and agile contexts.

### 2.1 Capability maturity models in the COTS context

The Software Engineering Institute (SEI) has developed the Software Acquisition Capability Maturity Model (SA-CMM) [14], which may be considered the major contribution of CMM to the COTS selection context. SA-CMM supports the acquisition process providing a maturity model to manage the organization acquisition processes in a highly disciplined manner. Consequently, with the most recent version of CMM, the CMMI product suite (and with the Acquisition Module CMMI-AM), it could be possible migrate from SA-CMM to CMMI [15]. In Table I, we describe some improvements from CMMI over SA-CMM, which are critical for defining,

building, fielding, and supporting COTS-based systems [16], and achieving the improvement over COTS selection processes.

**Table I. Improvements of CMMI over SA-CMM**

Process Area	Description of Innovation or Update
<b>Project Management</b>	
Project Planning	Takes into account the management and the maintenance of project information.
<b>Project Monitoring and Control</b>	
Risk Management	Suggests determining a administration strategy to manage the risk with the purpose of mitigating it and to avoid its negative impact.
Process and Product Quality Assurance	The evaluation criteria takes into account the organization goals when the evaluation is performed.
Supplier Agreement Management	The service and products could be identified sending the requirements to different providers
Integrated Supplier Management	This new area proposes to handle the relationship with the supplier and to prevent in the acquisition process anomalous situations.
Measurement and Analysis	Addresses the measures that capture information that allow the transition towards quantitative processes
<b>Engineering Concepts</b>	
Requirements Development	Considers the management of users who request components from the market
Requirements Management	The traceable management takes into account the requirements administration to work in parallel over the user requirements and the technical solution with the purpose of controlling the changes of requirements
Product Integration	Suggests monitoring acquired components whilst they are integrated with continuous validations and verifications
Decision Analysis and Resolution	Provides a set of guidelines to steer the analysis and the evaluation to take a suitable decision.
<b>Process Management Concepts</b>	
Organization Environment for Integration	Training of people is performed to promote a continuous collaboration where the organization vision could be shared by team members.
Integrated Project Management	Suggests a shared and coordinated vision between the selection team and the organization, to obtain the project goals.
Integrated Teaming	Defines and manages the set of roles that belong to the work team.
<b>Optimizing Concepts</b>	
Organizational Innovation and Deployment	Suggests the selection and the incremental development of the organization technology.

### 2.2 Agile methods in the COTS context

Currently, agile methods are starting to be considered in contexts other than traditional software engineering, for instance in product line engineering [17, 18]. Therefore it may be natural to analyze these methods in the COTS context, as we have previously done [11], and it is also done in [19] for the case of

ERP systems implementation, more focused on project management and implementation than in selection, which is natural due to the coarse granularity of ERP systems. In Table II we analyze the agile values in the COTS context to complement the agility analysis performed in [11]:

**Table II. Influence of agile values in COTS processes.**

Agile Values	Influences over COTS processes
“Individuals and interactions over processes and tools”	The information systems based in COTS components are used by the organizational staff, and they are who take the advantages offered by the system architecture. Therefore, in COTS projects it is very important to consider the human factor, because different actors participate in several selection processes where it is important to define a right way of communication, to collaborate continuously between the stakeholders, and to share knowledge from different disciplines.
“Working software over comprehensive documentation”	In COTS selection processes this value is not applied totally, because we do not work in software when we are selecting a COTS component from the marketplace. Maybe, this value has an important influence when we need to integrate the selected component to the system architecture because the development of glue-code to integrate a component requires an effort that exceeds the effort of developing code for in-house components. On the other hand, within the context of this proposal, this value can be addressed to work in selecting a suitable component, rather than documenting all the processes and events that happened during the project
“Customer collaboration over contract negotiation”	Agile methods take into account the relationships between the people who participate in software projects to improve the collaboration and communication between team members. In COTS projects it is also important to manage the relationships between team members that participate in the acquisition development, because there are different disciplines that need to share knowledge to steer the project successfully. On the other hand, we must consider external relationships from the organization with COTS providers to obtain a better assistance and customization over the products acquired.
“Responding to change over following a plan”	The COTS component marketplace and the information systems are in constant evolution. The marketplace evolves quickly by the competition pressures and by technological advances, forcing COTS providers to deliver new versions, updates and releases of COTS components. On the other hand, information systems are updated with a regular frequency, acquiring or licensing components from the marketplace. For these reasons, the system architecture and the selection team must be flexible to manage the continuous changes that could affect the COTS project.

These agile values can influence positively the COTS selection processes, and they can provide foundation to suggest agile practices to improve the agility in the COTS selection process.

### 3. Main processes in COTS Selection

In this section we identify the most relevant processes that appear during COTS selection. To perform this description, we have analyzed some of most widespread methods of COTS selection, among them we mention CARE [20], SCARLET [21], OTSO [22], EPIC [23] and STACE [24], with the purpose of identifying the main selection processes proposed by them. In Table III, we describe at high level the main processes involved in each of these selection methods. We may observe that the selection methods define different processes, or the same processes with different names, but in fact all of them are closely related. It is necessary then an effort to consolidate these proposals into a unifying one. First, for understandability purposes, we group these processes into five main areas (see Table III for the respective matching between COTS areas and processes of selection methods): *system architecture analysis, requirement engineering, market exploration, candidate component evaluation, and component selection*. Next, we look for an existing framework that may be used as a unifying one, as mentioned above. We use the OPEN Process Framework (OPF) [25], which defines a repository of classes that includes concepts bound to business modelling, business decision making, maintenance, application development, and COTS processes modelling. And then, we select processes from OPF, which are called “activities” that are in this framework that may be bound to the processes identified in the table III. We present the result in the rest of the section.

#### 3.1 System architecture analysis

Before integrating a component from the marketplace into a specific information system, we must consider the constraints, restrictions and composition of the system architecture. For this reason, we need a system description to know the main features over which we integrate the new components. In Table IV, we describe the five main tasks that take part of the system architecture process defined in OPF.

**Table III. High level processes of COTS selection methods.**

COTS Selection Methods	Main High Level Processes	COTS Areas
CARE	Define system agents	Requirement Engineering
	Define system goals (with COTS)	Requirement Engineering
	Define system requirements (with COTS)	Requirement Engineering
	Define engineering domain requirements (with COTS)	Market Exploration, Candidate Component Evaluation
	Define outline of architecture (with COTS)	System Architecture
BANKSEC	Acquire information from all participants	Requirement Engineering
	Analyse the acquired information	Requirement Engineering, Market Exploration
	Make decisions about component requirement compliance	System Architecture
	Select/reject candidates components	Candidate Component Evaluation, Component Selection
OTSO	Search	Requirement Engineering
	Screening	Market Exploration
	Evaluation	Candidate Component Evaluation
	Analysis	System Architecture
	Deployment	Candidate Component Evaluation, Component Selection
	Assessment	Candidate Component Evaluation, Component Selection
EPIC	Gather information	Market Exploration
	Refine into harmonized set	System Architecture
	Assemble executable	Requirement Engineering, System Architecture
	Assess iteration	Candidate Component Evaluation, Component Selection
STACE	Requirements definition	Requirement Engineering
	Alternatives/identification	Market Exploration
	Evaluation (assessment)	Candidate Component Evaluation
	Social-technical criteria definition	Component Selection

### 3.2 Requirement engineering

Requirements engineering applied in COTS projects, depends on the dynamics and evolution of the components available from the marketplace. The processes and technical tools that we use to steer the

elicitation and specification of requirements must try to adapt user needs to the real state of marketplace. In table V, we describe the main tasks involved in COTS requirement engineering.

**Table IV. System architecture tasks.**

Activities	Description
Architecture Reuse	It is focused in seeking the reusable elements and artefacts within the system architecture. In order to carry out this task, we need an architecture description timely updated.
Architecture Prototyping	Considering this task can help to support and verify the decisions that can impact over the system architecture
Architecture Production	This task identifies the set of features that compose the system architecture, determining their advantages, restrictions, and constraints against the user requirements.
Architecture Documentation	The purpose of this task is gathering the architecture information that we must store to learn about and search for the components that we can reuse, and the added functionality of the system components
Architecture Integrity Assurance	This task preserves the architecture integrity to be not violated when we select or we integrate a new component in the system architecture

**Table V. Requirement engineering tasks**

Activities	Description
Business Analysis	It specifies the set of tasks necessary to compare the user requirements against organizational goals. For this reason, the market analysis is performed, identifying the suitable providers, analyzing the specific market technology segment where the organization technology is developed. Furthermore, the user goals are analyzed throughout the COTS project to preserve the user vision.
Requirement Development	This task comprises the set of activities necessary to carry out the requirements formulation. Therefore, activities such as to identify properly the user requirements, to hold a continuous requirements analysis, and to negotiate the requirements with user representatives, are essential activities in COTS projects to maintain the system architecture integrity, to understand the user needs, and to seek a suitable component from the marketplace
Requirement Management	It includes the set of activities necessary to manage properly user requirements; for example, activities to negotiate, to store and to control requirements are proposed.
Vision	This requirements engineering task produces and documents the vision of user representatives about a required component.

### 3.3 Market exploration

The COTS marketplace is composed of different kinds of technology segments to acquire or license COTS components. Currently, we can find a significant

quantity of COTS information from the marketplace. Nevertheless, the dynamic and continuous updates of COTS components in the market makes the component information obsolete quickly. For this reason, we need selecting the necessary information produced in the market exploration process taking into account the provider information, the COTS component features themselves, and other aspects that we can reuse. In Table VI, we resume the main tasks in the market exploration process.

**Table VI. Market exploration tasks**

Activities	Description
Candidate Component Identification	Identifies the candidate components from the market, performing comparisons between the users requirements and the functionality offered by COTS vendors.
Candidate Component Solution Identification	Identifies the set of components from the marketplace. They conform the component solution.
Candidate Vendor Analysis	This task analyzes the main aspects of possible vendors, to evaluate them for their suitability to take part in a candidate solution.
Component Vendor Monitoring	Monitoring the COTS vendor helps us to establish relationships with mutual benefit in which we can work together with them.

### 3.4 Candidate component evaluation

We can find different components from the marketplace that can adjust to user requirements. The evaluation process must take into account techniques and tools that help to discriminate between the different component options. The team that steers component evaluation must have either knowledge or experience in the component domain under evaluation. Therefore, the team members must be able to handle technical tools; besides they must have a good understanding of the users' needs to evaluate components according to these needs. In Table VII, we describe the main tasks performed in candidate component evaluation.

**Table VII. Candidate component evaluation tasks**

Activities	Description
Candidate Solution Component Evaluation	Comprises the set of candidate components which are part of the possible final solution to be evaluated.
Candidate Component Evaluation	Its responsibility is evaluating the candidate component features.
Business Impact Analysis	This task analyzes the impact of component candidates over objectives of the business.
Candidate Vendor Analysis	It analyzes the main aspects of possible vendors, to support component evaluation.

### 3.5 Component selection

We need considering different criteria to choose a suitable component, because neither the most expensive component nor the cheapest one are necessarily the most advisable components to integrate into the information system. There are a lot of aspects that play a crucial role when selecting a candidate component, such as the contract, the component aggregated functionality, the verification of the functionality offered by the COTS vendor and the integration ability, among other factors. In Table VIII, we describe briefly the main tasks that are part of selection processes.

**Table VIII. Component selection tasks**

Activities	Descripción
Component Vendor Monitoring	Represents the set of tasks to control and monitor the vendor representative in the selection processes and component integration.
Component Monitoring	Undertakes the tasks to control the component performance during the evaluation processes and selection. Besides, it proposes the control of future versions and releases of selected components.
Business Process Modification	Its responsibility is measuring the impact of the selected component over the business goals.
Component Selection	Comprises the set of tasks needed to discard or select the candidate component that is part of the possible final solution
Update Definition of System Architecture	This definition is relevant to maintain updated the changes of the system architecture when we select any COTS component to be integrated.

Finally, in Table IX we analyze the degree of coverage of COTS selection methods over the processes areas and selection activities presented in this section, with the aim of assessing the adequacy of these methods. For this purpose, we provide a rationale for this analysis where:

- A mark 'C' means that the process area is explicitly covered by COTS selection method.
- A mark 'P' means the process area is partially covered by COTS selection method.

## 4 Reconciling the agile and CMMI contexts

In this section, we suggest a balance that shows a situation that helps us to determine a reconciliation point among the necessary discipline to develop COTS projects and the necessary agility to carry out selection processes, so we can define "what we can do" and "how we can do it" to develop successfully the COTS

**Table IX. Coverage of selection areas by selection methods.**

Activities	COTS Selection Methods				
System Architecture Analysis Process Area					
	CARE	BANKSEC	OTSO	EPIC	STACE
Architecture Reuse	C	C	C	C	P
Prototyping	P	P	P	P	P
Architecture Production	C	C	C	C	P
Arch. Integrity Assurance	C	C	P	C	P
Architecture Documentation	C	C	C	C	P
Requirement Engineering Process Area					
	CARE	BANKSEC	OTSO	EPIC	STACE
Business Analysis	P	P	P	C	C
Requirement Development	C	C	C	C	C
Requirement Management	C	C	C	C	C
Vision	C	C	C	C	C
Marketplace Exploration Process Area					
	CARE	BANKSEC	OTSO	EPIC	STACE
Candidate Component Identification	C	C	C	C	C
Candidate Com. Solution Identification	C	C	C	C	C
Candidate Vendor Analysis	C	P	C	C	C
Component Vendor Monitoring	P	P	P	C	P
Candidates Components Evaluation Process Area					
	CARE	BANKSEC	OTSO	EPIC	STACE
Candidate Solution Com. Evaluation	C	C	C	C	C
Candidate Component Evaluation	C	C	C	C	C
Business Impact Analysis	C	P	P	P	C
Candidate Vendor Analysis	C	C	C	C	C
Component Selection Process Area					
	CARE	BANKSEC	OTSO	EPIC	STACE
Component Vendor Monitoring	P	P	P	C	P
Component Monitoring	P	C	P	C	P
Business Process Modification	P	P	P	P	P
Component Selection	C	C	C	C	C
Definition of Architecture	C	C	C	C	C

the set of new activities and roles of COTS selection [4, 26]. For this reason, recognizing CMM models in COTS projects could help us to achieve a strategic discipline to address the processes improvement in COTS domain. On the other hand, in COTS discipline there are some important aspects that could be supported by agile methods, among others: the need for flexibility in user requirements definition [2, 4, 5, 8]; sharing knowledge between different kinds of disciplines that must work together in selection [3 - 6]; considering that human factors in COTS selection processes can help to improve project management, collaboration between stakeholders, and technical excellence [27, 11]. The reason of this are that the agile approaches have practices based on time-boxed iteration, evolutionary development, adaptive planning, evolutionary delivery, and inclusion of other values and practices that encourage agility in the software development context. Currently, these two contexts have generated controversy [28], and a wide debate is also carried out on the controversy of agile foundations. For instance, some important subjects of this discussion are: the tacit knowledge [29, 30]; innovation of agile methods [28]; and misconceptions about agile methods [30, 31, 32]. Besides, there are some specific subjects about CMM models that must be considered: in the CMM context people working in project development should make an effort to practice and to achieve skills which will be institutionalized by the organization, forcing them not to pay attention on the tasks and needs of the project, but on the objectives and practices that have not been carried out yet; in the process development we need to have many candidates practices rather than bureaucratic and fixed practices [33]; or some authors point out that the CMMI model help us to manage the bureaucracy and boilerplate with its emphasis on risk management and integrated teaming [34]. Beyond these subjects, some authors are seeking the right way to work together with CMMs and Agile contexts, where it is possible to take advantage of two contexts. For example, Paulk analyzes XP from a CMM perspective in [35]: he highlights the discipline and effectiveness of some XP practices. Also, Boehm and Turner suggest the identification of 5 critical dimensions (size, criticality, dynamism, personnel, and culture) that can be used to describe an organization or a project in terms of its agile and plan-driven characteristics [36]. In Figure I, we sketch our proposal to put together disciplined processes (using CMMI), and best agile practices to drive COTS selection processes. This agreement point suggests a balance among the agility and discipline that may be achieved through improvement which is provided over selection

process areas. This reconciliation point seeks being complemented with two important dimensions that influence any development methodology which takes into account the system criticality of COTS selection processes (such as: lost of comfort, lost of discretionary money, lost of essential money, lost of lives) and the number of people that play a role the COTS project [37].

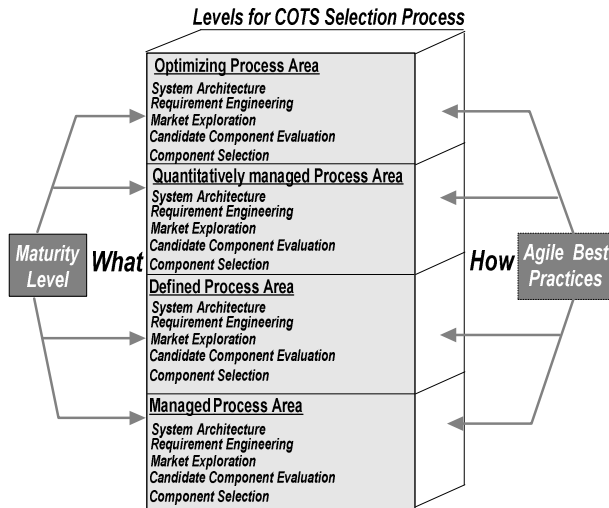


Figure I. Maturity levels and best practices applied to COTS selection processes.

With these two dimensions we would be able to regulate the necessary discipline inside COTS selection processes, using practices that may be adjusted to the specific needs of the COTS projects, because the number of people that participate inside the COTS project and the criticality involved by COTS projects help us to apply more or less discipline depending on selection processes ceremony that we need to apply. On the other hand, the formality and discipline that should be applied during the selection of a specific component varies according to component criticality and according to its impact over process ceremony. For example, if we consider integrating two new components into an information system, one for the financial management of data, and another for the management of the internal organization news, the necessary degree of ceremony to acquire these components can vary, according to the number of people that participate during the selection and according to processes criticality. In Figure II, we may observe that the tool for financial management (represented with the black box) requires for its selection and integration between 7 - 20 people due to its criticality level. On the other hand, the tool for news management (represented with the grey box) needs less personal because it implies a smaller effort and a

smaller criticality level than the financial tool. We can evaluate with this identification over which tool we need more planning, more qualified personal and less ceremony to develop COTS selection processes.

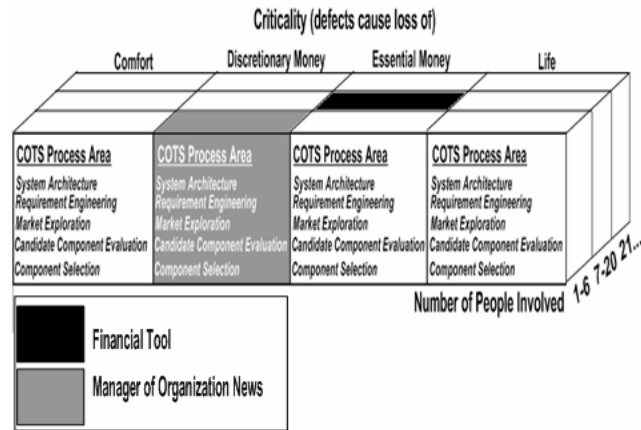


Figure II. Maturity levels and best practices applied COTS selection practice areas, based on [37].

## 5 The Framework

In this section we present the framework that supports the agreement point to steer COTS selection processes based on CMMI and agile contexts (see table X). This framework is made up of three main domains:

- The first domain is the CMMI context that specifies what we can do to obtain disciplined COTS selection processes. We use from this domain its stage representation [38]. The stage representation is structurally composed of five maturity levels to predict the performance of an organization, improving its internal processes. These maturity levels are made up of process areas, which are a set of related activities that are performed together to achieve the specific and generic goals:
  - Maturity Level 1 (initial): in this first level there are neither processes nor activities defined for being used in selection projects. Selection processes are usually ad hoc and chaotic. Suitable COTS selection depends on the ability of the organization members that perform the selection.
  - Maturity level 2 (managed): this level manages the selection processes to steer the COTS projects according to their documentation plans. It ensures that requirements are managed and determine which processes are planned to be performed, measured, and controlled.
  - Maturity Level 3 (defined): this level takes into account the standardized and defined selection

processes to be applied over different COTS projects. These standard processes are established consistently across the organization, and they are managed proactively using an understanding of the process activities relationships and detailed measures of the process.

- Maturity Level 4 (quantitatively managed): the main feature of this level is providing quantitative objectives for quality over selection process performance. These objectives are based on the needs of the customer, end users, organization, and process implementers.
- Maturity Level 5 (optimizing): this level focuses on continuous improvement of selection process performance throughout the information system life-cycle. Improvement processes are made up of incremental and innovative technological improvements.

In Table X we present an excerpt of our framework, namely the second level.

- The second domain that influences our framework is the agile context, because it provides the set of best practices based on agile values, which specifies how we can obtain agility over selection processes. We have analyzed some agile methods to study their best practices with the purpose extrapolating them toward COTS context, as XP [39], SCRUM [40], Crystal Methods [41], and FDD [42]. The role of these best practices is to help to achieve the specific and generic goals defined by each process area. These practices could be applied over any COTS project taking into account the criticality and the number of people involved to perform the selection. In table X, we put in the second column the initial set of best practices that are able to be extrapolated from agile methods towards the COTS context. Furthermore, they are able to influence the CMMI processes areas to complement them, and regulating the discipline that we need when we develop the selection processes.
- COTS selection processes areas is the last domain involved in this framework. These areas involved in any COTS project are able to take advantage from these two previous domains to steer suitably the selection processes. In Table X we have represented the matching between the CMMI process areas, and the practice areas that would be applied over the COTS context. The COTS process areas are able to be influenced by agile practices to achieve the specific or generic goals defined by each CMMI process areas, taking into account the system criticality of a COTS project and the number the people involved to perform it.

**Table X. Framework proposed**

WHAT CAN WE DO (CMMI)	COTS Selection Processes Areas						HOW CAN WE DO IT (Agile)
	System Architecture	Requirement Engineering	Market Exploration	Evaluation	Candidates Components	Component Selection	
<b>MATURITY LEVEL 2: MANAGED</b>							
Requirement Management	Metaphor (XP) Product Backlog (SCRUM) Domain Object Modeling (FDD)						
Project Planning	Planning game (XP) Pre-game planning and staging (SCRUM) Staging (Crystal)						
Project Monitoring and Control	Small releases (XP) Pair Programming (XP) Collective Ownership (XP) Monitoring (SCRUM) Revision and Control (Crystal) Inspections (FDD)						
Supplier Agreement Management	Sprint Review Meeting (SCRUM) Feature teams (FDD)						
Measurement and Analysis	Planning game (XP) Effort estimation (SCRUM) Developing by feature (FDD)						
Process and Product Quality Assurance	Testing (XP) On-site Customer (XP) Methodology tuning technique (Crystal) Progress reporting (FDD)						

With these three domains working together we are able to manage the negative impact of lessons learned introduced in previous section (see section 1), because the CMMI and agile context are made up of fundamental features which offer coverage in: responding quickly over requirements changes; these context take into account the relevant system users to work together with them to understand and comprehend their needs; their foundations allow addressing the organization in a commercial manner to change either their policy or their business processes. In Table XI we present the influence of one specific CMMI process area over COTS selection process areas to explain our agreement point. The CMMI process area is *Requirement Management Processes Area*, its main purpose is managing the project requirements identifying inconsistencies between those requirements and the project's plans and work products [38]. This process area has a direct influence over three specific COTS process areas specifying what we can do:

- Over COTS system architecture: we must manage the dependencies and inconsistencies among the system architecture and requirements (see section 3.1).
- Over COTS requirements engineering: the component requirements must be managed to control the requirements changes (see section 3.2).



- Over COTS market exploration: the user requirements depend of marketplace situation, for this reason, the requirements have to be adapted to the component market (see section 3.3).

**Table XI. Framework proposed**

CMMI Requirement Management Process Area		COTS Processes Areas Influenced			
Goal Process Area	Practices by Goal Process Area	Architecture System	Requirement Engineering	Marketplace Exploration	
Manage Requirement Specific Goal	<ul style="list-style-type: none"> <li>- Understanding of requirements.</li> <li>- Commitment to Requirements.</li> <li>- Manage Requirements Changes</li> <li>- Bidirectional Traceability of Requirements</li> <li>- Identify Inconsistencies between Project</li> </ul>	<ul style="list-style-type: none"> <li>- Metaphor (XP)</li> <li>- Product Backlog (SCRUM)</li> <li>- Domain Object Modelling (FDD)</li> </ul>			Agile Practices

Moreover, the *Requirement Management Process Area* defines two objectives that we need to fulfill in order to address the characteristics that describe what must be implemented to satisfy the process area:

- Manage requirements (specific goal): the requirements are managed and inconsistencies are identified [38].
- Institutionalize a managed process (generic goal): it establishes and maintains an organizational policy for planning and performing the requirements management process [38].

In addition to these goals, CMMI defines for each process area a set of practices to describe what activities are expected to result in the achievement of its goals. In the *Practices by Goal Process Area* column of Table XI NO EXISTE, we identify the practices by each goal of *Requirement Management Process Area*. Furthermore, in Table XI NO EXISTE we have selected some agile practices (from XP, SCRUM, Crystal Methods, and FDD) to be extrapolated over COTS process areas. These agile practices are able to specify how we can achieve the goals defined:

- Agile practices for *Manage Requirements* goal:
  - Metaphor (XP): specifies how we can understand the user requirements where all stakeholders are able to learn and describe the component required with a common idea about it.

- Product backlog (SCRUM): it defines the work to be done in the process area, where multiple stakeholders are able to participate describing the requirements and the features to achieve them.
- Domain object modeling (FDD): it is focusing on modeling and describing the domain of the problem.
- Agile practices by *Institutionalize a Managed Process* goal:
  - Planning game (XP): estimates the effort needed for the exploration of COTS requirements in the marketplace deciding the scope for market exploration.
  - Pre-game planning and staging (SCRUM): plans the meetings with the stakeholders to define the features of the selection project with the purpose generating work for first iteration.
  - Staging (Crystal): defines the schedule for iteration where the selection team selects the requirements to be implemented in the next increment.

With these elements we are able to regulate the discipline that we may apply over some COTS project. For example, if we consider selecting a *manager of organization news* tool, the system criticality for these kinds of tools represents the lost of discretionary money, and likely we need few people involved in the selection processes. On the other hand, if we need selecting a *financial tool*, we must be more careful at the moment to apply these agile practice because the system criticality for these kinds of tools represent the loose of essential money, for this reason we need more ceremony to develop the selection processes.

## 6 Conclusions

The processes involved in COTS projects require activities and roles to apply coordination, discipline, and commonality, besides it is necessary to share the knowledge generated among different disciplines that participate to attain a suitable component selection to satisfy the user needs. This study analyzes the influence of agile and CMMI contexts over COTS selection processes, which have generated controversy inside the software engineering community, with the purpose of suggesting an agreement point of reconciliation and balance among the necessary discipline required to develop a selection process, and the agility that we are able to provide to develop a COTS project. For this reason, we seek to take advantage of the discipline proposed in CMMI, and the agility of the best agile practices, to identify a point of balance that define the

number of people involved in the COTS selection process development and the system criticality that should be having into account at the moment to carry out a COTS project. These contexts can be applied over selection development in a suitable or unsuitable way, having into account the need of ceremony or formality that are required in COTS selection processes, helping us to define what we can do and how we can do it to obtain a component from the marketplace in a suitable way. On the other side, with this agreement we suggest a research starting point.

## References

- [1] Brownsword, L., Oberndorf, T., Sledge, C. "Developing New Processes for COTS-Based Systems". *IEEE Software* (Vol. 17, No. 4) July/August 2000 pp. 48-55.
- [2] FAA SERC. *Lessons Learned in Developing Commercial Off-The-Shelf (COTS) Intensive Software System* 2000. Available: <http://www.faa.gov/aio/common/documents/HTMLfiles/LssLrnd.htm>
- [3] Brownsword, L., Place, P. "Lessons Learned Applying Commercial Off-the-Shelf Products" CMU/SEI-99-TN-015, 2000.
- [4] Morisio, M., et al. "COTS-based software development: processes and open issues". *Journal of Systems and Software* 61 (2002): 189-199.
- [5] Office of the Secretary of Defense (OSD). *Commercial Item Acquisition: Considerations and Lessons Learned 2000*. Available: [http://www.acq.osd.mil/dpap\\_archive/Docs/cotsreport.pdf](http://www.acq.osd.mil/dpap_archive/Docs/cotsreport.pdf)
- [6] Maiden, N., Ncube, C., Moore, A. "Lessons Learned During Requirements Acquisition for COTS Systems" *Communication ACM* 40(12): 21-25 1997.
- [7] Adams, R., Eslinger, S. "Best Practices for the Acquisition of COTS-Based Systems: Lessons Learned from the Space System Domain" in *Proc. ICCBSS 2004: 203-205*.
- [8] Ncube, C., Maiden, N. "Selecting COTS Anti-Virus Software for an International Bank: Some Lessons Learned" in *Proc. 1<sup>st</sup> MPEC Workshop, 2004*.
- [9] Reifer, D., Basili, V., Boehm, B., Clark, B. "COTS-Based Systems - Twelve Lessons Learned about Maintenance" in *Proc. ICCBSS 2004: 137-145*
- [10] Torchiano, M., Morisio, M. "Overlooked Aspects of COTS-Based Development". *IEEE Software* 21(2), 2004.
- [11] Navarrete, F., Botella, P., Franch, X. "How Agile COTS Selection Methods are (and can be)?" in *Proc. Euromicro 2005*. Porto, Portugal.
- [12] Paulk, M., Curtis, B., Chrissis, M., Weber, C. "Capability Maturity Model SM for Software Version 1.1", Technical Report CMU/SEI-93-TR-024, ESC-TR-93-177, February 1993.
- [13] Clements, P., Northrop, L. *Software Product Lines*, Addison-Wesley, ISBN 0-201-70332-7, 2002.
- [14] Cooper, J., Fisher, M. "Software Acquisition Capability Maturity Model (SA-CMM<sup>®</sup>)", Technical report CMU/SEI-2002-TR-010 Version 1.03 March 2002.
- [15] Fisher, M., Goethert, W., Jones, L. "Applying the Software Acquisition Capability Maturity Model". *CrossTalk Journal*, Aug. 2002, Vol. 15 No. 8, pp. 4-7.
- [16] Tyson, B., Albert, C., Brownsword, L. "Implications of Using the Capability Maturity Model Integration (CMMi©) for COTS-Based Systems" in *Proc. of the second international conference on COTS-Based Software Systems*, p.p., 229 – 239, 2003.
- [17] Kurmann, R. "Agile SPL-SCM Agile Software Product Line Configuration and Release Management". in *Proc. of 1<sup>st</sup> International Workshop on Agile Product Line Engineering (APLE'06)*. Maryland, USA. 2006.
- [18] Carbon, R., Lindvall, M., Muthig, D., Costa, P. "Integrating Product Line Engineering and Agile Methods: Flexible Design Up-front vs. Incremental Design". in *Proc. of 1<sup>st</sup> International Workshop on Agile Product Line Engineering (APLE'06)*. Maryland, USA. 2006.
- [19] Alleman, G. "Agile Project Management Methods for ERP: How to Apply Agile Processes to Complex COTS Projects and Live to Tell About It". in *Proc. XP/Agile Universe LNCS 2418*, 2002.
- [20] Chung, L., Cooper, K., Courtney, S. "COTS-Aware Requirements Engineering and Software Architecting". in *Proc. of the SERP* 2004.
- [21] Maiden, N., Kim, H., Ncube, C. "Rethinking Process Guidance for Selecting Software Components". in *Proc. of 1<sup>st</sup> ICCBSS*, LNCS 2255, 2002.
- [22] Kontio, J. "A Case Study in Applying a Systematic Method for COTS Selection". in *Proc. of 18<sup>th</sup> Intl' ICSE*, 1996.
- [23] Albert, C., Brownsword, L. "Evolutionary Process for Integrating COTS-Based System (EPIC): An Overview". Report CMU/SEI-2002-TR-099 ESC-TR-2002-009, July 2002.
- [24] Kunda, D. "STACE: Social Technical Approach to COTS Software Evaluation". *Component-Based Software Quality - Methods and Techniques*, LNCS 2693, 2003.
- [25] Firesmith, D., Henderson-Sellers, B., Graham, I. *OPEN Modeling Language (OML) Reference Manual*. Cambridge Univ. Press, New York, 1998.
- [26] Boehm, B., Abts, C. "COTS integration: plug and pray?" *Computer Volume 32, Issue 1, Jan. 1999* Page(s):135-138.
- [27] Tate, K. *Sustainable Software Development: An Agile Perspective*. Addison Wesley Professional ISBN: 0-321-28608-1 October 11, 2005.
- [28] Berard, E. "Misconceptions of the Agile Zealots". Report the Object Agency, Available: <http://www.svspin.org/Events/Presentations/MisconceptionsArticle20030827.pdf>, 2003. (Last Accessed April 2006).
- [29] Boehm, B. "Get ready for Agile Methods, with care". *Computer (IEEE)*, pp. 64-69, January 2002.
- [30] McBreen, P. *Questioning Extreme Programming*. Addison Wesley, 2003

- [31] Jeffries, R. "Misconceptions about XP" an Agile Software Development Resource January 2002. Available: <http://www.xprogramming.com/xpmag/Misconceptions.htm> (Last Accessed January 2006)
- [32] Fowler, M. "Is design dead?" XP2000 Proceedings. Available: <http://www.martinfowler.com/articles/designDead.html>. Last Significant Update: May 2004 (Last Accessed April 2006).
- [33] DeMarco, T., Lister, T. *Peopleware—Productive Projects and Teams*, 2nd Ed., Dorset House, 1999
- [34] DeMarco, T., Boehm, B. "The agile Methods Fray". IEEE Computer, Vol. 35, No. 6, June 2002, pp. 90-92.
- [35] Paulk, M., "Extreme Programming from a CMM Perspective". IEEE Software 18(6): 19-26 (2001)
- [36] Boehm, B., Turner, R. "Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods". in Proc. of the 26th International Conference on Software Engineering (ICSE'04).
- [37] Cockburn, A. *Agile Software Development*. Addison Wesley 2000-2001.
- [38] CMMI Team. *Capability Maturity Model® Integration (CMMISM), Version 1.1*. Technical report CMU/SEI-2002-TR-011, ESC-TR-2002-011. March 2002
- [39] Beck, K. *Extreme Programming Explained: Embrace Change*, Addison Wesley, 1999.
- [40] Schwaber, K. "The Scrum development process". in *Proc. OOPSLA '95 Workshop on Business Object Design and Implementation*, Austin, 1995.
- [41] Cockburn, A., Crystal Clear. A human-powered methodology for small teams, including The Seven Properties of Effective Software Projects, Addison Wesley 2002.
- [42] Palmer, S., Felsing, J. *A Practical Guide to Feature-Driven Development*. Prentice-Hall 2002.