

Evaluation and Selection COTS Software Process: The State of the Art

Feras Tarawneh¹, Fauziah Baharom², Jamaiah Hj. Yahaya³, and Faudziah Ahmad⁴

^{1,2,4}College of Arts and Sciences, Universiti Utara Malaysia (UUM)

Sintok, Kedah, Malaysia

¹feras79tara@hotmail.com

²fauziah@uum.edu.my

⁴fudz@uum.edu.my

³Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia

Bangi 43650, Selangor, Malaysia

³jhyahaya@yahoo.com

ABSTRACT

In the recent years, the Commercial Off-The-Shelf (COTS) products are being increasingly used in the world of software development. Therefore, evaluating and selecting appropriate COTS product is one of the most critical activities in COTS-based system development. Unfortunately, many methods that have been proposed in previous studies for evaluating and selecting COTS software are still have many limitations and lack to apply and accept as a formal method in the industry. So without an effective method for evaluating and selecting COTS products, the time spent on selecting the suitable COTS software may offset the advantages of using it. This paper outlines and discusses the common problems in existing methods and presents the main processes and evaluation criteria (non-functional requirements) that are required for evaluating and selecting COTS software through theoretical and empirical studies which goal is to develop new framework to evaluate and select COTS software.

KEYWORDS

COTS software; COTS evaluation and selection; theoretical framework; COTS mismatches; evaluation criteria.

1 INTRODUCTION

The world of software development has changed a lot in recent years, the software functionalities have become

more complex because the rapidly changing in the customers' demands and the software technology in the market is evolved very fast. Therefore, a new approach has been produced as an alternative software development approach which is based on integrating pre-packaged solutions, usually known as Commercial-Off-The-Shelf (COTS) software.

COTS software is defined as commercially pieces of reused software that are developed and supported by outside suppliers (so-called vendors) to integrate and re-use by other software projects to provide additional functionalities within a final system. There are several COTS software products have been successfully applied in the software development such as office automation software (word processor, calendars, spreadsheet, etc.), messaging system, database, operating systems, and geographic information system (GIS).

COTS-based system is developed based on selecting, adapting, and integrating one or more COTS software, this process is also called COTS-Based System Development (CBSD) [1]. CBSD is the act of composition. This approach changes the way of building software development from in house-development to pre-existing COTS software that has

been tested many times by many other users. Thus, this approach grants the opportunity to lower the costs, time and effort for developing systems. Also CBSD enhances the reliability, flexibility, and reusability of the systems [2].

On the other hand, several challenges are encountered by the organizations when using COTS software for developing their systems. One of main challenges is the lack of abilities to select the most suitable COTS software that meets their requirements. This challenge occurs due to many similar COTS software in the market with different capabilities and qualities characteristics [2]. In contrast, any wrong decision for selecting COTS software will reflect negatively on the project as entire by increasing of the cost, time, effort, and also effect negatively on the performance and quality of the final system [3]. Therefore, most of the organizations are interested in the evaluation and selection process and considered it as one of the critical success factors in CBSD [4], [5].

However, the evaluation and selection COTS software process has many problems such as rapid evolvement of the COTS software market [6], the “Black box” nature of COTS software, evolving requirements during COTS evaluation [7], ineffective evaluation criteria, and lack of well-defined and systematic COTS software evaluation and selection process [8].

This paper investigates the common problems and limitations of existing methods for evaluating and selecting COTS software, presents general objectives of the work, and the methodology for proposing new framework to evaluate and select COTS software. This framework will be developed based on theoretical and empirical studies. In additional, the paper

presents the theoretical framework for evaluating and selecting COTS software.

2 COMMON PROBLEMS OF THE EVALUATION AND SELECTION COTS SOFTWARE

COTS software evaluation and selection process is the most important process in the CBSD. Therefore many models have been proposed to handle the COTS software evaluation and selection problems, but all of these models have a lack to be accepted and considered as formal method for evaluating and selecting COTS software in industry [3]. The evaluation and selection of COTS software is still performed using ad-hoc manners [9], such as depending on the experiences of developer team or their intuition, or depending on the relationship with particular vendor. Therefore, lack in systematic, repeatable, and well-defined process for evaluating and selecting COTS software in the industry keeps the organizations under the pressure [10]. Furthermore, the development team has lack of experiences to plan for the selection process in detail [4].

Even though of many methods have been proposed previously to evaluate and select COTS software, there are some issues and problems that are still not considered by these methods such as lack of identifying the mismatches between user requirements and COTS features, lack of handling non-functional requirements to distinguish between COTS software alternatives, and lack of managing and learning from previous selection cases knowledge.

2.1 Mismatches Between COTS Software Features and User Requirements

The main development approaches that are used to develop CBS are: buy the COTS software and used (that means no

mismatches), and buy the COTS software and adapting it to fit the current system and requirements (that means there are mismatches between COTS software and user requirements) [4]. Therefore, team developers should consider all of users' requirements or expectations and system architecture from side, and COTS products capabilities in the other side. Typically, the integrating of the COTS product that is developed to meet general needs overall market with local system that has specific characteristics and requirements will suffer from many mismatches. These mismatches can be classified into two types: Architectural mismatches, and COTS mismatch [11], [12].

2.1.1 Architectural Mismatches

The mismatch of architectural emerges when the COTS software components do not match with the system or other components in the system. Typically, the architectural mismatches arise from interoperability and incompatibilities

with other parts of the system such as using different database schema, different programming languages, or communication protocols [13]. However, the architectural mismatches have little impact on the COTS-solution system and intermediate CBS, while they clearly appear in the COTS-intensive system.

2.1.2 COTS software Mismatches

This kind of mismatches appears between COTS software and user requirements or expectations. These mismatches arise from developing COTS software for wide use while user requirements have specific characteristics or needs [14]. Mismatches are defined as a shortages or excesses of COTS features against customer requirements [4]. However, the COTS mismatches were classified into several categories; Table1 shows the types of COTS mismatches as mentioned by [15].

Table1. Types of COTS mismatches

Mismatches Types		Explanation	Example
Differ		This kind of mismatches means there are partial matching between user requirements and COTS software.	"The new software should save file in DOC and PDF format file" but the COTS software does not support the PDF format.
Fail		This kind the COTS software completely fails to achieve the requirement.	"The new software should save the modifications automatically" but the COTS software does not support these requirements.
Extend	Helpful	It means the extra feature of the COTS software is accepted.	The COTS software allows user to edit text with grammars checking as extra feature.
	Hurtful	It means the extra features have negative impact on the system.	COTS software allows to automatic data backup, this facility can affect the performance of system.
	Neutral	The extra features do not meet with achievement of any user requirements nor its desired functionality.	"The COTS software allows user to import XLS files (Excel worksheet)" but this feature doesn't requested by the user and also doesn't important to him.

Determining the COTS mismatches and the types of these mismatches are not enough to identify the fitness COTS software. Therefore, it's required to

know how to solve the COTS mismatches and what the resources (cost, time, effort) are needed to solve them, and the risks that resulted by applying resolutions [4], [15]. There are

two scenarios can be used to solve the COTS mismatches:

- Requirements modification: is an attempting to customise the requirements to meet COTS software features.
- Adapting COTS software: at this scenario, the resolving mismatches are conducted on the COTS software features.

However, adapting COTS software to user requirements can be done by such Add-ons, API (Application Programming Interface), and Scripting language [4], [16].

Identifying the mismatches between COTS features and customer requirements does an important role for supporting the decision making in COTS software selection [1], [17]. Thus, early addressing and better understanding these mismatches will support and provide valuable insight on the decision of COTS software selection and thereby, reduces the risk of project failure. Also most of these mismatches are solved after selecting the COTS software which makes the latter activities like adaptation and integration in CBSD easier [17], [4]. In reality, most of existing methods for evaluating and selecting COTS software neglect the mismatches between COTS features and customer requirements [18].

2.2 Non-Functional Requirements (NFRs) Problem

As a practical and theoretical of requirements engineering have lack of dealing with NFRs, the current methods for evaluating and selecting COTS software also have lack on handling NFRs [19]. Typically, many empirical reports indicated that there is lack or incorrect dealing with NFRs which cases failure, delays, or increases the final cost and effort of projects [20]. However, NFRs are defined as a general quality

attributes, specifications, and the constraints on the software product [17], [18]. Kassab et al in [20] defined NFRs as soft goals that have no clear description or standards to verify if it has been achieved or not.

However, NFRs are considered as advance and critical criteria in the evaluation and selection COTS software process. Precisely, NFRs play important role to distinguish similarities of COTS software alternative and facilitate the COTS software evaluation and selection [19]. NFRs are related to the software as complete characteristics rather than the individual characteristics, and often might be a deciding factor on the survival of software. So they define as the overall characteristics or attributes of the system such as quality attributes, vendor attributes, organizations attributes [23]. NFRs for COTS software evaluation have been classified by [19] into four groups: quality attributes, architectural attributes, domain attributes, and organization attributes. Table 2 shows the brief description and examples of these groups.

Analysis of NFRs helps and supports the decision makers through the selection process, and NFRs like operability and adaptability help for selecting suitable COTS software that will be easily integrated with particular system architecture [24]. Nevertheless, NFRs may be difficult to elicit, express, quantify and test, also it being a relative with specific domain, and interact with each other, and subjectively when they are interpreter and evaluated [24].

The common limitation of current methods of COTS software selection is more concerned on functionality requirements and cost criteria over the non-functional requirements [19]. Conversely, most existing methods for evaluating and selecting COTS software

do not provide sufficient support and [19].
 clear definition about these requirements

Table2. Types of NFRs

NFRs Types		Description	Example
Quality attributes		The quality attributes provide the means for measuring the fitness and suitability of a product.	reliability, usability, maintainability, and performance
Architecture attributes	Independence	The independence makes components to be easy replaced by component from other resources	integrity, portability, reusability, flexibility, evolvability and scalability
	cooperation	The cooperation supports communication between different software components in the system	interoperability, and composability
Domain attributes		these requirements are derived from the application domain not from the users and they often reflect fundamentals of the application domain	majority (e.g. compliance to domain standard), researched (e.g. popularity of the COTS software in particular domain)
Organizational attributes	User organization	there are many constrains or attributes should be known about the user organization before acquiring COTS software	characteristic of existing hardware platform, legacy application kind, current environment of software development
	Vendor organization	A set of attributes that are required by the users on the vendor organization before integrating their products	Vendor stability on the market, vendor credentials, the strategy of supporting the product.

2.3 Lack of Learning from Previous COTS Software Selection Cases Knowledge

Previous software selection cases provide an important knowledge that supports and helps both of the evaluators and decision makers in current software selection. It is necessary to review and study the information from previous software evaluation and selection cases. Such of this information:

- The previous software components that were chosen.
- Successful criteria and techniques that were used before beginning the current evaluation and selection process.
- The information about set of vendor attributes that will be very important in current selection case, such as vendor reputation, vendor sustainability, and vendor credibility [25].

However, most methods of evaluating and selecting COTS software in the literature recommended documentation of COTS software evaluation and selection process. These methods do not showing the mechanism of storing, and how to manage the information during COTS software

evaluation and selection process in order to learn from the previous selection cases and support the decision making process [3]. So, the important issue about learning from previous evaluation and selection cases is how to store, access, and retrieved the data and information in easy and effective method. According to Wanyama and Far in [26], the information repositories have important role in the management of COTS evaluation and selection information, and they provide the best mechanism to store and retrieve the information of the COTS software evaluation and selection process in effective manner. But developing and accessing these repositories has taken place with a little attention in the previous studies [3].

3 GENERAL OBJECTIVES

This paper presents one part of general research that aims to propose a new framework to support and improve the COTS software evaluation and selection processes in industry. To achieve this objective several specific objectives have been addressed: (1) identifying the processes that are support the COTS software evaluation and selection; (2) to determine the criteria or requirements are

required for successful evaluation and selection process; (3) to propose suitable method and technique to address the mismatches between COTS features and customer requirements; (4) to develop a simple repository to manage information from previous selection cases that will support the decision making process.

4 LITERATURE REVIEW

Set of elements should be considered when developing new framework for COTS software evaluating and selecting. Such of these elements is the main processes and factors (evaluation criteria) that supporting COTS software evaluation and selection. Also the common methods for evaluating and selecting COTS software will be taken into account when building the framework.

4.1 Evaluation Criteria

Identifying the evaluation criteria is very important task for understanding, evaluating, and selecting the suitable COTS software. Evaluation criteria are decomposed through the evaluation criteria definition in a hierarchical decomposition, which starting from high level requirements until producing pieces of well-defined measurement information. Evaluation criteria are defined based on careful analysis of many influencing factors such as application requirements, application architecture, project objectives and constraints (budget and schedule) [27]. However, there is lack of providing a general list of evaluation criteria that can be used to evaluate and select COTS software. Therefore, the evaluation criteria must include functional requirements (what the software expected to do, or the services of software) and non-functional requirements (performance, interface, maintenance and support, cost, etc.) [19].

Several characteristics should be considered when defining the evaluation criteria, such of these characteristics [28]:

1. Good criteria for evaluation and selection COTS software should be included a clearly measurable statement and measurement method.
2. The criteria can be a good if the data that is collected support the measurement method.
3. The ability to discriminate is a very important aspect for the good criteria to help the evaluators to distinguish between COTS software.
4. Good criteria should be a non-overlapping because if the criteria are overlapping, may it leads for conflicting in analyzing the COTS software features, misleading results, and wasted effort.
5. If the criterion is not valuable in context of system, then it must not be used (e.g. long-term stability of vendor is not irrelevant if the COTS product will be used as a short-term).

However, There are a different sources of decomposing criteria like the organizations check lists and software literature, the most common sources are ISO and IEEE standards, for example ISO include a set of software quality attributes (e.g. functionality, reliability, usability, etc.) and suggest set of measurement methods. Moreover, many techniques have been used for representing and defining the evaluation criteria, the common techniques have been used by existing methods are:

- a) Goal Question Metric (GQM): this technique is the common technique that is used for defining the evaluation criteria by refinement the requirements into capability statements and measurement methods. the GQM technique consists of [28]:
 1. The goal, which is to achieve the requirement.
 2. The question, which is a capability statement.
 3. The metric, which is the standard of measurement.

Table3 presents an example how to define the evaluation criteria by GQM technique.

Table 3. Example of GQM Technique

Goal (Requirement)	Minimum impact of learning curve on end-user performance.
Questions (Capability statement)	<ul style="list-style-type: none"> • Are the text and graphics visible? • Is there an intuitive interface? • Are the training materials effective?
Metrics (Measurement standard)	<ul style="list-style-type: none"> • Readability of data by user with aging eyesight • Reaction of user to a sample set of common tasks • Retention of information via a formal exam

(Adopted from [28]).

b) Non-functional requirements framework: NFRs framework is used for representing and analyzing NFRs, and understanding of what a particular NFRs means. However, NFRs framework represents the high-level requirement as soft goals, and then these soft goals are decomposed into sub-goals (more specific). In the NFRs framework, NFRs can be contributed as positively or negatively, and fully or practically to meeting other NFRs. Also, NFRs framework includes two kinds of contribution: i) AND relationship that mean if the soft goal decomposed into more specific sub-goals, then all of these sub-goals should be together achieved the high-level soft goal, ii) OR relationship that means the soft goals are satisfied by any of its sub-goals [29].

Figure1 presents an example for decomposing the soft goal security of information into sub-goal like integrity, availability, and confidentiality through AND kind of contribution, also shows how the cryptograph contributes negatively (appears as“-“) for system performance [29].

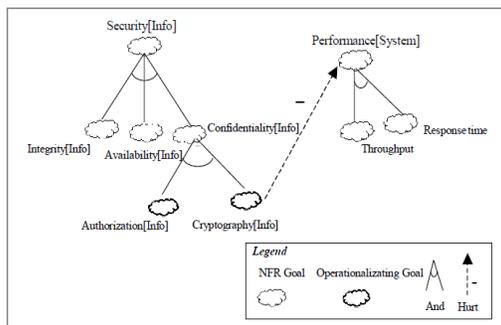


Figure 1. Decomposition of NFRs Using the NFR Framework (adopted from [29]).

4.2 Main Processes for Evaluating and Selecting COTS Software

Based on previous studies, several processes for evaluating and selecting COTS software are shared by existing methods for COTS software selecting. These processes can be ordered as iteratively, sequentially, or overlapping. However, the common processes for evaluating and selecting COTS software can be classified in terms of four general processes [30]:

4.2.1 Supporting process

This process consists of set of activities that support other processes of the valuation and selection. This process begins with planning for an evaluation and selection COTS software, the tasks that might be completed during this activity include forming the evaluation and selection team (e.g. technical experts, domain experts, end users, etc.), identifying stakeholders (e.g. integrators, (funding customers, business owners, etc.), define the goals and objectives, etc [18], [31]. Documentation is also performed during this process. Document the recommendations from the evaluation team are done in several forms like COTS software dossier and evaluation record. COTS software dossier includes several kinds of information that is needed to understand and use the COTS software product, also contain of information about vendor, limitations of COTS software, discovered facts, also if the COTS software is rejected it will show the rejection reasons [28].

4.2.2 Preparation Process

The main purpose of this process is to collect and prepare the information that required for further detail evaluation. Therefore, this process begins with establishing the evaluation criteria (include functional and non-functional criteria) and in

order to identifying the potential COTS software candidates, searching activity is performed depending on the initial criteria which called searching criteria. Searching activity requires the use different resources of information for searching the COTS software (e.g. internet, in-house components libraries, magazine and journal vendors, trade show and conferences, publication and sales promotions), because depending on a single source restricts the search space [32]. However, the output of the searching activity is often too general and very large list of COTS software alternatives, thus the screening activity (filtering) is performed to decrease this list of alternatives to reasonable number for further detail evaluation [31].

4.2.3 Evaluation Process

Each of the evaluation criteria definition and identifying the COTS software alternatives have important role on the evaluation success [32]. This process plays a vital role to determine how well each of the COTS software alternatives achieves the evaluation criteria [18]. Therefore, the main objective of this process is to estimate each COTS software alternative against the evaluation criteria in more detail and sort these alternatives based on their importance. Several techniques can be used to consolidate the evaluation data such as Analytic Hierarchy Process (AHP) and Weighted Scoring Method (WSM) for decision making in the selection process [27].

4.2.4 Selection Process

The outputs of the evaluation process are several kinds of data such as facts, checklists, weights, opinions. Those kinds of data should be consolidated and interpreted into information [18]. However, the decision maker requires knowing about previous selection cases, and identifying mismatches between COTS features and customer requirements in order to select the fitness COTS software. The recommendations in the last to the manager should include either

buying COTS software products or building the software in house [31], [27].

4.3 COTS Evaluation Strategies

Practically, there are three strategies are founded in the previous studies for evaluating the COTS software: progressive filtering, keystone, and puzzle strategy.

4.3.1 Progressive Filtering Strategy

This strategy begins with large number of COTS software candidates in the list, and then each potential COTS software candidate is met with set of discriminating criteria which are defined through successive iteration of COTS software estimation cycle [10], [31]. COTS software that does not satisfy these evaluation criteria is progressively removed from the COTS software candidates list in each cycle of estimating. This strategy is done iteratively until the fitness of COTS software candidates are identified and remained in the list, then selecting one or more of COTS software can be done from the list for integrating in the application [24].

4.3.2 Keystone Strategy

At this strategy the COTS software candidates are estimated against a key characteristic [10]. So the key characteristics (e.g. vendor location, type of technology) are identified at the beginning of this strategy, then the searching for COTS software will based on the satisfying this keystone characteristic. This strategy is applied at the beginning stages of the evaluation in order to permit quickly removing the large number of COTS candidates that do not satisfy the keystone characteristic [31].

4.3.3 Puzzle assembly strategy

The idea of this strategy is taken from collecting pieces of a puzzle [18]. This strategy assumes that when selecting the COTS software must consider the fitting of the COTS software with other components

on the system [10], [24]. In other words, COTS software that can be considered as fitness in isolation, it might be not acceptable when assembled with other components in the system. Therefore, at this strategy, choosing COTS software must be done with consider the other components requirements in the puzzle.

Mohmad et al in [18] argues that more than strategies from above can be used with the same project, such as the keystone can be used at the beginning of the project to eliminate the largest possible number of COTS candidates, and then the progressive filtering can be used later on.

4.4 Existing Methods for Evaluating and Selecting COTS Software

Many methods have been carried out dealing with evaluation and selection COTS software in the previous studies. These methods can be clustered into requirements-driven approaches represented by Procurement-Oriented Requirements Engineering (PORE) [5] and COTS-based Requirements Engineering (CRE) [24], while Off-The-Shelf-Option framework (OTSO) [27] and Social-Technical Approach to COTS software Evaluation (STACE) [32] represented the architecture-driven approaches [3].

4.4.1 OTSO

Off-The-Shelf Option (OTSO) [27] is the first widespread method for evaluating and selecting COTS software. It supports many techniques which are used for determining the evaluation criteria, cost and benefits estimation of candidates, and supports decision making like Analytical Hierarchy Process technique (AHP) [33]. OTSO method is considered as an important milestone and basis model for the other methods. However, OTSO method has several limitations such as (1) lack of considering non-functional requirements like vendor aspects which consider the functional

and cost aspects; (2) it doesn't provide specific technique about how to handle the extra or unrequired features (mismatches problem); (3) it also depended on AHP technique to provide the decision making although this technique has several limitations like it not efficiently in large number of comparisons.

4.4.2 PORE

Procurement-Oriented Requirements Engineering (PORE) method [5] is a template-based method to select COTS software. It is based on an iterative process of requirements elicitation and product selection. PORE method integrates set of techniques, methods and tools, such as: multi-criteria decision making techniques, knowledge engineering techniques, and requirements acquisition techniques. Also PORE method offers guidelines for designing product evaluation test cases. Conversely, PORE is not clear in specifying requirements and eliminating the products (i.e. do not capture the decision rationale). PORE based on templates to acquire and evaluate COTS alternatives, but these templates provide only initial view of steps to do a systematic evaluation.

4.4.3 STACE

Social-Technical Approach to COTS software Evaluation (STACE) [32] was developed to address the lack of attention in non-technical issues for COTS software like organization issues and social issues. On the other hand, the main limitation of this method is the lack on a process of requirements acquisition and specification. Moreover, it is not clear how to deal with mismatches problem, also this approach does not provide or use systematic analysis of COTS alternatives during the assessment when using a decision-making technique.

4.4.4 CRE

COTS-based Requirements Engineering (CRE) [24] is an iterative COTS software

selection approach that chooses COTS software by rejection. CRE consider time restriction, domain coverage, vendor guaranties, and cost rating through the evaluation process. However, CRE approach does the balance between the evaluated cost and benefits without any guidance that explain how to satisfy it. Also this approach has a lack of supporting experiences and information sharing between stakeholders. Furthermore, the decision will be more complex and a large number of final situations as a resulting for dealing with large number of COTS alternatives. CRE has less ability to handle COTS software selection and it is most suitable for requirements elicitation.

In general, despite the similarities between these methods by sharing several processes, factors, and techniques, there are missing issues are not considered and still not addressed by these approaches such as identifying mismatches between COTS features and customer requirements. In additional, those methods are concern on functionality and cost criteria over the non-functional criteria. Also, these methods have limitations to provide suitable and systematic mechanism to manage and learn from previous selection cases in order to support the decision making.

5 METHODOLOGY

The main objective of this work is to propose a new framework for supporting and improving COTS software evaluation and selection processes in software industry. In order to do so, it is necessary to identify set of processes and evaluation criteria that support COTS software evaluation and selection.

This research follows the deductive approach [34], because this approach begins with general idea (such as theory, principles, and concepts) and moving to more specific conclusion, this also called “top-down” approach. It is suitable to be applied in

developing a model, where theories or concepts will be derived from theoretical and empirical findings. Then the proposed model will be applied and evaluated in real environment. This methodology consists of four stages: (1) stat-of-the-art study, (2) stat-of-the-practice study, (3) framework development, and (4) framework evaluating.

5.1 State-Of-The-Art Study

In this phase, previous studies will be reviewed in depth and focused on several related topics like such as COTS-based systems, evaluation and selection process, existing methods for evaluating and selecting COTS software. The main aim of this phase is to identify and analyze the common processes and evaluation criteria that have been used in evaluating and selecting COTS software. Additionally, the deep analysis on the existing methods and models for evaluating and selecting COTS software will be carried out. Also the related issues to evaluation and selection COTS software such as evaluation strategies, mechanisms, guidance, and templates that facilitate implementing evaluation and selection process will be investigated in this study.

The deliverables of this study will be a set of theoretical processes and factors for evaluation and selection COTS software, and common limitation of existing methods and models. Moreover, the questionnaire that is required in the next phase will be designed and tested using pilot study.

5.2 State-Of-The-Practice Study

The overall purpose of this study is to investigate the current practices of COTS software evaluation and selection which are related to the use of process, factors, and relevance issues (mechanisms, templates). It aims to determine the importance of the current theoretical processes and factors related to the evaluation and selection COTS software in practice. It is important to understand the current evaluation and

selection situation in practice and the problems that are faced by the organizations.

In this phase, self-administered questionnaires will be used because it has several advantages such as cost effectiveness, ease to analysis, coverage a wide area, and it supports a high degree of secrecy [35]. The survey will be conducted in Jordan where the respondents are from IT organizations (that have experience with CBS) including the IT manager, developers, and other software practitioners. The data collected will be coded and entered to Statistical Package for Social Science (SPSS) software for analyzing it.

5.3 Framework Development

In this stage, the findings from theoretical and empirical studies such as the successful processes, factors, and related issues (templates, guidance, and mechanisms) to COTS software evaluation and selection will be used for developing a new framework. The development of new framework for evaluating and selecting COTS software aims to bridge the gap between state-of-the-art and state-of-the-practice. However, the proposed framework will be constructed by integrating set of processes and their activities, also the relationship between them will be established. The successful factors (NFRs) for evaluating and selecting COTS software will be determined and established. Also many techniques will be used for eliciting the user requirements, and for identifying the information about COTS software alternatives, and also the technique for determine mismatches between COTS features and customer requirements. Moreover, this framework will be supported by a simple repository tool in order to control and learn from previous selection cases in order to support the current decision making process. The proposed framework will be also supported by set of guidance, mechanism, evaluation strategies and templates to facilitate evaluation and

selection COTS software and support CBSD in the real life.

5.4 Framework Evaluating

The aim of this phase is to evaluate the effectiveness and acceptability of the proposed framework in real environment. The evaluation will facilitate improvement and refinement to the proposed framework. A case study will be adopted as a qualitative method because this method is preferred when the researcher cannot control or manipulate the relevant behavioural events [36]. The evaluation process will start by determining the criteria that will be used to evaluate the framework. However, interviews will be adopted as a data collection method because its flexibility and adaptability, open-end questions will be used among the interviews in order to void the interview bias. The data will be entered into a software tool for analysis (e.g. ATALAS/it) and the modification and refinement will be conducted if required.

6 THEORETICAL FRAMEWORK

Based on literature review, we propose a theoretical framework for evaluating and selecting COTS software which include two studies: theoretical study (state-of-the-art), empirical study (state-of-the-practice). Theoretical study focuses on three main issues: processes (activities and techniques), evaluation criteria (non-functional requirements), and previous frameworks for selecting COTS software. The empirical study will use survey and case study to investigate the elements from theoretical study in the real life. Figure 2 shows how these studies are used to achieve the aims of this research.

The theoretical framework shows the main issues that should be considered when developing a new framework for evaluating and selecting COTS software. One of these issues is a set of processes that should be followed to select more fitness COTS

software. This diagram shows all the main processes that have not been included by the most previous methods like preparing process, and supporting process. In this research, the selection process will be more focused because the final decision about selecting COTS software is prepared at this process [31]. The decision makers face many challenges to decide the suitable COTS product. The main challenge is how to identify the mismatches between the COTS software features and customer requirements to select the most fitness COTS software with minimum cost and effort to adapt and integrate with other components. Moreover, Learning from the previous COTS software evaluation and selection cases helps the evaluators and decision makers to understand how the past software components were

chosen and which successful criteria and techniques were used. Known about past selection cases contributes to supports greatly the experiences of evolution and selection team [4], [25].

The evaluation criteria play a vital role during the evaluation and selection COTS software. As the theoretical framework shown, the evaluation criteria are classified into functional and non-functional criteria [23]. The non-functional criteria are considered vital because they play important role to distinguish between the COTS software such as quality attribute (reliability and efficiency), domain attributes (maturity and security), architectural attributes (portability and integrity), and organization attributes (vendor attributes) [19].

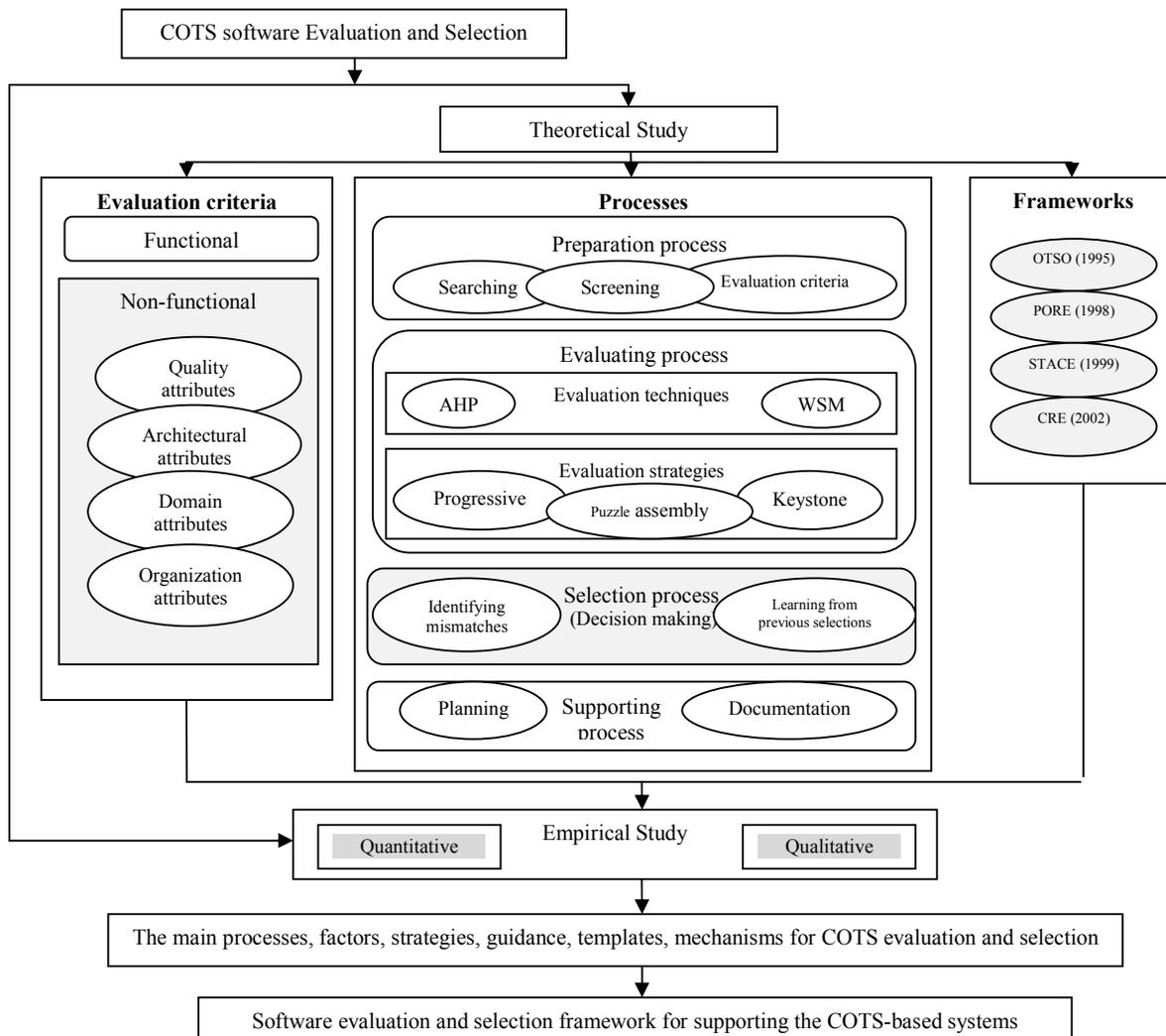


Figure 2. The proposed theoretical framework

On the other hand, for eliciting and synthesise current practices of COTS evaluation and selection the empirical study will be conducted based on theoretical study by using quantitative study (questionnaire) and qualitative study (case study) to get the successful processes, criteria, techniques, strategies, and mechanisms for building a new framework for evaluating and selecting COTS software.

7 CONCLUSION AND FUTURE WORK

In this paper we presented theoretical study (state of the art) of COTS software evaluation and selection. This study has investigated the common problems and limitations of the previous studies on the evaluation and selection COTS software; classified the main processes and criteria that are required for evaluating and selecting COTS software, and explained several strategies for conducting evaluation process. In addition, a well-defined methodology for developing a new framework for evaluating and selecting COTS software was presented in this paper. The theoretical framework of this research has also been presented. Our next step is to investigate the state of the practice by conducting survey on the set of organizations that applying CBSD in order to investigate the processes and criteria from theoretical, and identifying the current practices of the evaluation and selection of COTS software. According to the Findings from this survey as well the findings from theoretical study, a new framework for evaluating and selecting COTS software will be developed to support CBSD.

8 REFERENCES

1. A. Kvale, J. Li, and, R. Conradi, "A Case Study on Building COTS-Based System Using Aspect-Oriented Programming," in Proceedings of the 2005 ACM symposium on Applied computing, Santa Fe, New Mexico, pp. 1491 – 1498, 2005.
2. T. Wanyama and B. Far, "An Empirical Study To Compare Three Methods For Selecting Cots Software Components," International Journal of Computing and ICT Research, vol. 2, p. 34, 2008.
3. T. Neubauer and C. Stummer, "Interactive Decision Support for Multi Objective COTS Selection," in System Sciences, HICSS 2007 40th Annual Hawaii International Conference on, pp. 283b-283b, 2007.
4. A. Mohamed, G. Ruhe, and A. Eberlein, "Optimized Mismatch Resolution for COTS Selection," Software Process Improvement and Practice, vol. 13, p. 157, 2008.
5. N. Maiden and C. Ncube, "Acquiring COTS Software Selection Requirements," IEEE software, vol. 15, pp. 46-56, 1998.
6. P. Ulkuniemi and V. Seppanen, "COTS Component Acquisition in an Emerging Market," IEEE software, vol. 21, pp. 76-82, 2004.
7. K. Meinke, "A Stochastic Theory of Black-Box Software Testing," Algebra, Meaning and Computation, pp. 578-595, 2006.
8. K. Vijayalakshmi, N. Ramaraj, and R. Amuthakkannan, "Improvement of Component Selection Process Using Genetic Algorithm for Component-Based Software Development," International Journal of Information Systems and Change Management, vol. 3, pp. 63-80, 2008.
9. C. Coutts and P. Gerdes, "Integrating COTS Software: Lessons from a Large Healthcare Organization," IT Professional, vol. 12, pp. 50-58, 2010.
10. D. Kunda, "STACE: Social Technical Approach to COTS Software Evaluation," Lecture Notes in Computer Science, pp. 64-84, 2003.
11. P. Avgeriou and N. Guelfi, "Resolving Architectural Mismatches of COTS Through Architectural Reconciliation " in The 4th International Conference on COTS-Based Software Systems (ICCBSS'05), Bilbao, Spain, 2005, pp. 248-257.
12. M. Shaw, "Architectural Issues in Software Reuse: It's Not Just the Functionality, It's the Packaging," in ACM SIGSOFT Symposium on Software Reusability, 1995, pp. 3-6.
13. E. Mancebo and A. Andrews, "A Strategy for Selecting Multiple Components," in Proceedings of the 2005 ACM symposium on Applied computing, Santa Fe, New Mexico, 2005, pp. 1505 - 1510.

14. D. Carney, S. A. Hissam, and D. Plakosh, "Complex COTS-Based Software Systems: Practical Steps for Their Maintenance," *Journal of Software Maintenance*, vol. 12 (6), 2000 pp. 357-376.
15. Alves, C., Franch, X., Carvalho, J.P., Finkelstein, A.: "Using Goals and Quality Models to Support the Matching Analysis During COTS Selection", *Proc. of ICCBSS 2005*, LNCS 3412, 146-156, 2005.
16. M. R. Vigder, W. M. Gentleman, and J. Dean, "COTS Software Integration: State of the art," *National Research Council Canada (NRC)*, 39198, Jan 1996.
17. C. Alves and A. Finkelstein, "Investigating Conflicts in COTS Decision-Making," *International Journal of Software Engineering and Knowledge Engineering*, vol. 13, pp. 473-493, 2003.
18. A. Mohamed, G. Ruhe, A. Eberlein. "COTS Selection: Past, Present, and Future". In *Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)*, Tucson, Arizona (USA), 2007.
19. L. Beus-Dukic, "Non-Functional Requirements for COTS Software Components," in *Proceedings of ICSE workshop on COTS Software*, pp. 4-5, 2000.
20. M. Kassab, M. Daneva, and O. Ormandjieva, "Early Quantitative Assessment of Non-Functional Requirements," *Centre for Telemetric and Information Technology*, University of Twente, Enschede, Technical Report TR-CTIT-07-35, Citeseer, 2007.
21. N. Ernst, Y. Yu, & J. Mylopoulos, "Visualizing Non-Functional Requirements," Paper presented at the *Proceedings of the 1st international workshop on Requirements Engineering Visualization*, Canada, pp. 2-2, 2006.
22. I. Sommerville, "Software Engineering," 6th ed. Addison- Wesley, 2001.
23. C. Pavlovski and J. Zou, "Non-Functional Requirements in Business Process Modelling," in *5th Asia-Pacific Conference on Conceptual Modelling (APCCM2008)*, Wollongong, NSW, Australia, pp. 103-112, 2008.
24. C. Alves and J. Castro, "CRE: A Systematic Method for COTS Components Selection," in *XV Brazilian Symposium on Software Engineering (SBES)*, ed: Rio de Janeiro, Brazil, 2001.
25. A. Alghamdi, "An Empirical Process for Evaluating and Selecting AWEM Environments: Evaluation Stage," vol. 19, pp. 17-37, 2007.
26. T. Wanyama and H. B. Far, "Repositories for COTSSC Selection," In the *proceedings of the Canadian Conference on Electrical and Computer Engineering. CCECE 2006*, Ottawa, Ontario, Canada, May 1-3, 2006.
27. J. Kontio, "OTSO: A Systematic Process for Reusable Software Component Selection," *University of Maryland at College Park*, College Park, MD, 1995.
28. S. Comella-Dorda, J. C. Dean, G. Lewis, E. Morris, P. Oberndorf, and E. Harper, "A Process for COTS Software Product Evaluation," *CMU Univ., SEI Institute*, CMU/SEI-2003-TR-017, July 2004.
29. R. Hill, J. Wang, K. Nahrstedt, "Quantifying Non-Functional Requirements: A Process Oriented Approach," in the *Proceedings of the 12th IEEE International Requirements Engineering Conference*, Kyoto, Japan, September 2004.
30. Land R., Blankers L., "Classifying and Consolidating Software Component Selection Methods," *MRTC report ISSN 1404-3041 ISRN MDH-MRTC-218/2007-1-SE*, 2007.
31. R. Land, L. Blankers, M. Chaudron, and I. Crnkovic, "COTS Selection Best Practices in Literature and in Industry," in *Proceedings of 10th International Conference on software Reuse*, Beijing, China, pp. 100-111, 2008.
32. D. Kunda and L. Brooks, "Applying Social-Technical Approach for COTS Selection," in *Proceedings of the 4th UKAIS Conference*, 1999.
33. T. Saaty, "The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation," *MacGraw-Hill*, New York International Book Company, p. 287, 1980.
34. W.M.K. Trochim, "Deduction & Induction Thinking," Available <http://www.socialresearchmethods.net/kb/dedind.php>, Retrieved 10 April 2010.
35. J. Kirakowski, "Questionnaires in Usability Engineering," Available at www.ucc.ie/hfrg/resources/qfaq1.html, Retrieved 5 April 2010.
36. R. Yin, "Case Study Research," in *Design and Methods*, 3rd, London: Sage, 2003.