COTS Selection: Past, Present, and Future

Abdallah Mohamed¹, Guenther Ruhe¹, Armin Eberlein²
¹University of Calgary, 2500 University Drive, NW, Calgary, AB, T2N1N4, Canada
²American University of Sharjah, Sharjah, P.O. Box 26666, UAE
{asamoham, ruhe, eberlein}@ucalgary.ca

Abstract

Commercial Off-The-Shelf (COTS) products are increasingly being used in software development. In COTS-based development, selecting appropriate COTS is the most crucial phase. This paper explores the evolution of COTS selection practices, and surveys eighteen of the most significant COTS selection approaches. The paper traces how each approach contributed to the improvement of current COTS selection practices, and then compares them. The paper also highlights some open research issues relevant to the selection process, and concludes with a discussion of possible future directions to address these issues.

1. Introduction

Most of today’s software systems include one or more COTS (commercial off-the-shelf) products. COTS are pieces of software that can be reused by software projects to build new systems [1, 2]. Such COTS include word processors, email packages, etc [3]. Performing a good COTS selection plays a critical role in the success of the final system [4]. COTS selection is the process of determining the fitness-of-use of COTS products in a new context, and then selecting one or more products with the highest fitness [5].

COTS selection involves many challenges such as the high complexity of the process [6]. To overcome these challenges, several efforts were made during the last decade to model the COTS selection process. However, none of these methods can be considered as the silver-bullet to solving the COTS selection problem. Different approaches are of different effectiveness and might be suitable for different contexts.

In this paper, we show how different COTS selection approaches contributed to the advancement of COTS selection practices. The paper reviews current COTS selection approaches, discusses their contribution, compares them, shows their pros and cons, and briefly explains their main activities. Next, the paper highlights those issues that were not sufficiently addressed by existing approaches, and shows possible future directions that can be used to solve these issues. To the best of our knowledge, there is no previous such thorough surveys.

This paper is structured as follows: Section 2 describes the COTS selection process. The section starts by describing the so-called “general COTS selection process” which highlights the main activities that most existing approaches use. Then, we discuss the evaluation process which is the core of any COTS selection approach. Section 3 presents COTS selection approaches and summarizes their contribution to the evolution of COTS selection practices. Section 4 presents some of the ‘present’ open issues relevant to the COTS selection process. Then, in section 5 some ‘future’ directions that can be taken to address these issues are discussed. Finally, conclusions are given in section 6. The paper includes one appendix which describes the approaches listed in section 3.

2. The COTS selection process

2.1. The general COTS selection process

Although there is no commonly accepted method for COTS selection [6], all methods share some key steps that can be iterative and overlapping. These steps formulate what we refer to as the General COTS Selection (GCS) process which is described as follows:

Step 1: Define the evaluation criteria based on stakeholders’ requirements and constraints.

Step 2: Search for COTS products.

Step 3: Filter the search results based on a set of ‘must-have’ requirements. This results in defining a short list of most promising COTS candidates which are to be evaluated in more detail.

Step 4: Evaluate COTS candidates on the short list

Step 5: Analyze the evaluation data (i.e. the output of Step 4) and select the COTS product that has the best fitness with the criteria. Usually, decision making techniques, e.g. analytic hierarchy process (AHP) [7], are used for making the selection.

After Step 5, the selected COTS product is usually customized (aka tailored) as needed in order to reduce the mismatches it has with the requirements. A COTS product can be customized in different ways, such as using add-ons, adjusting parameters, etc.

2.2. COTS evaluation

COTS evaluation is the core of the COTS selection process as it is the activity in which the fitness-of-use
of COTS products is determined. COTS evaluation provides information necessary for the decision maker to select the best COTS product from a set of competing alternatives [8, 9].

COTS are evaluated against a set of criteria that represents the stakeholders’ requirements and system constraints. Kontio et al. [10] suggest to hierarchically define the evaluation criteria, where a set of high-level goals are gradually refined based on such factors as the application requirements and architecture, the COTS capabilities, etc. Maiden et al. [4] agrees with Kontio and further suggest defining the evaluation criteria while at the same time evaluating existing COTS. The technical literature also includes efforts to explain how to define the evaluation criteria based on quality models. For example, Franch et al. [11] and Carvallo et al [12] propose a six-step method to build a structured quality model for the purpose of COTS evaluation. They rely on the ISO/IEC 9126 quality model [13] and further explain activities to define a set of metrics that can be used during the evaluation.

Generally, there are three strategies which can be applied to evaluate COTS products [14, 15].

1. **Progressive filtering**, which starts with a large number of COTS, and then progressively defines discriminating criteria through successive iterations of product evaluation cycles, where in each cycle ‘less fit’ products are eliminated. This strategy requires running steps 1 to 4 in the GCS process iteratively until a small number of most promising COTS products is identified from which one or more can be selected for integration into the system.

2. **Puzzle assembly**, which assumes that a COTS-based system requires fitting various components together like pieces of a puzzle. This implies that a product that ‘fits’ in isolation might not be acceptable when combined with other products. Therefore, this strategy suggests considering the requirements of each product while simultaneously remembering the requirements of other products in the puzzle.

3. **Keystone identification**, which starts by identifying a key requirement (e.g. vendor location or type of technology), and then searching for products that satisfy this keystone requirement. This allows quick elimination of a large number of products that do not satisfy the key requirement.

More than one of the above strategies may be employed in the same project [15]. For example, a developer might use ‘keystone identification’ first and then later ‘progressive filtering’.

## 3. COTS selection approaches

This section surveys and compares current COTS selection approaches. The section tries to trace how each approach contributed to the improvement of current COTS selection practices. Each approach is discussed in more detail in Appendix 1.

### 3.1. The evolution of COTS selection practices

Figure 1 traces the improvements made to the COTS selection process over the last decade. One of the first proposals was given by Kontio et al. [16] who proposed the OTSO (Off-The-Shelf Option) approach for COTS selection in 1995. OTSO is considered an important milestone in the evolution of COTS selection practices as it served as a basis for other approaches. OTSO defined the basic structure of the COTS selection process. This structure is very similar to the GCS process described in section 2.1.

![Figure 1 Evolution of COTS selection practices](image-url)

In 1996, Kontio published several follow-up papers to elaborate OTSO (e.g. [10]) in which the process of defining the evaluation criteria is described in detail. In 1997, several approaches were proposed:

(i) The IusWare (IUStitia softWARis) [17] approach tried to formalize the selection process, and to address quality requirements during the evaluation process.

(ii) The PRISM (Portable, Reusable, Integrated, Software Modules) [18] approach proposed a generic component architecture that can be used during the COTS evaluation process.

(iii) The CISD (COTS-based Integrated Systems Development) [19] approach proposed a model that can be used to select multiple homogeneous COTS products.

However, it was not until 1998 that another important milestone was reached with the PORE approach [4]. The importance of PORE is that it proposed a requirements engineering process for COTS-based development. PORE suggested that requirements should be elicited and analyzed at the same time when the COTS products are evaluated.

In 1999, several approaches were proposed:

(i) The CEP (Comparative Evaluation Process) approach introduced the use of the so-called confidence factor (CF). The more reliable the source of data, the higher a CF value that source gets. Any estimate we make should be adjusted based on the CF value of the source based on which these estimations are made.
(ii) The STACE (Social-Technical Approach to COTS Evaluation) approach [20] emphasized the importance of non-technical issues, e.g. social, human, and organizational characteristics, during the evaluation process. (iii) The CRE (COTS-based RE) approach emphasized the importance of non-functional requirements (NFR) as decisive criteria when comparing COTS alternatives. CRE uses the NFR framework [21] to model NFRs. In 2000, Ochs et al. [22] proposed the COTS acquisition process (CAP) which emphasized the concept of a "tailorable evaluation process". This means the evaluation process (including the evaluation criteria themselves) should be tailored based on the available effort for the project. Ochs’ approach relied on experts’ knowledge to tailor the process.

In 2001, a project was started by Chung et al. to define a COTS-Aware Requirements Engineering (CARE) approach [23-26]. CARE uses a flexible set of requirements based on different agents’ views. For this, CARE proposes a method to define relevant agents as well as the system goals and requirements.

In 2002, another set of approaches were proposed: (i) the PECA (Plan, Establish, Collect, and Analyze) approach [27] from SEI [5] describes a detailed tailorable COTS selection process and gives guidelines which the experts can use to tailor the process. (ii) The BAREMO approach explained in detail how a decision can be made based on the AHP [7] method. (iii) The storyboard approach [28] suggests to incorporate use-cases and screen-captures during the requirements engineering phase to help customers understand their requirements, and thus acquire more appropriate COTS products. (iv) The combined selection approach [29] tries to select multiple COTS that are evaluated, first on the local scale to evaluate each COTS in isolation from the others, and then on the global scale to select the best combination of COTS.

In 2003, two more approaches were proposed: (i) The WinWin spiral model [3] which is a risk-driven approach that suggests to identify, analyze and resolve the risks in an iterative evaluation process. (ii) The approach by Erol et al. [30] suggests the use of fuzzy theory to quantify qualitative data, and then to use optimization techniques to determine optimal (or near optimal) solutions.

In 2004, the DesCOTS system [31] was presented based on the work done in [11]. This system integrates several tools to define evaluation criteria using quality models such as ISO/IEC9126 [13].

In 2005, the MiHOS (Mismatch-Handling aware COTS Selection) approach was developed [32]. MiHOS relies on the GCS method and introduces a process for handling mismatches between COTS products and requirements. MiHOS uses techniques such as linear programming to identify near optimal solutions.

3.2. Surveying COTS selection approaches

Appendix 1 describes the eighteen approaches mentioned in section 3.1 in detail. Table 1 compares these approaches in terms of the following criteria:

<table>
<thead>
<tr>
<th>APPROACH</th>
<th>COMPARISON FACTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GCS</td>
</tr>
<tr>
<td>A1 OTSO</td>
<td>95/96</td>
</tr>
<tr>
<td>A2 IusWare</td>
<td>1997</td>
</tr>
<tr>
<td>A3 PRISM</td>
<td>1997</td>
</tr>
<tr>
<td>A4 CISD</td>
<td>1997</td>
</tr>
<tr>
<td>A5 PORE</td>
<td>1998</td>
</tr>
<tr>
<td>A6 CEP</td>
<td>1999</td>
</tr>
<tr>
<td>A7 STACE</td>
<td>1999</td>
</tr>
<tr>
<td>A8 CRE</td>
<td>1999</td>
</tr>
<tr>
<td>A9 CAP</td>
<td>2000</td>
</tr>
<tr>
<td>A10 CARE</td>
<td>2001</td>
</tr>
<tr>
<td>A11 PECA</td>
<td>2002</td>
</tr>
<tr>
<td>A12 BAREMO</td>
<td>2002</td>
</tr>
<tr>
<td>A13 Storyboard</td>
<td>2002</td>
</tr>
<tr>
<td>A14 CS</td>
<td>2002</td>
</tr>
<tr>
<td>A15 WinWin</td>
<td>2003</td>
</tr>
<tr>
<td>A16 Erol's</td>
<td>2003</td>
</tr>
<tr>
<td>A17 DesCOTS</td>
<td>2004</td>
</tr>
<tr>
<td>A18 MiHOS</td>
<td>2005</td>
</tr>
</tbody>
</table>

- PF: Progressive filtering ✓ fully satisfies the criterion
- KS: Keystone ✓ does not satisfy the criterion
- PZ: Puzzle assembly ✓ partially informally or implicitly satisfies the criterion

3.3. Comparison

In this section, we list some of the open research issues that have not been sufficiently addressed by existing COTS selection approaches.

As we have seen in section 3, there is a great variety of approaches to tackle the COTS selection problem. Practitioners have two main requests: (i) they are wondering about the effectiveness of the different approaches, and (ii) they need guidance on when and how to choose these approaches. In his paper [1], R. Glass stated a clear message from software practitioners to software researchers: "What help do practitioners need? We need some better advice on how and when to use methodologies". This brings us to the first open research issue.
 Researchers may use these tools to build a more robust negotiation component that can address conflicts between stakeholders. This leads to the fifth issue:

R5: “To provide a negotiation component for resolving conflicts between stakeholders”

The remaining issues are related to the COTS market and their vendors. Most current approaches assume that the knowledge required for evaluating COTS products is available and reliable. Current approaches (e.g., [4]) suggest that COTS evaluation should initially rely on vendors’ documentations. Then, for specific requirements, we should rely on vendor-led demonstration and on user-led experimentation. Nevertheless, the problem here is three-fold: (i) searching for and collecting the knowledge might very effort consuming. This leads to another open research issue:

R6: “To design knowledge repositories and show how they can be used for evaluating COTS”

(ii) Some vendors are more reliable than others, and hence their documents can be used during the evaluation more broadly in order to reduce the evaluation effort related to user-led experimentations. This leads to another open research issue:

R7: “To develop techniques to evaluate vendors’ credibility, and adjust the COTS evaluation results (obtained from vendors’ documentations) based on such credibility estimation”.

5. An outlook

The seven issues described in section 4 can be analyzed for improving COTS selection practices. In this section, we show possible directions that can be used to address these eight issues.

For R1: A high-level roadmap was given in [1] to support selecting appropriate methods for different contexts. The main idea is to taxonomize existing methods, to categorize problem domains in terms of what they need, and finally to find a way to map these two categories to each other. In addition, further confirmative studies should be conducted to analyze the effectiveness of the wide variety of approaches.

For R2: Initial efforts [34, 35] developed a framework for customizing the COTS selection process. The main idea is to define a set of possible options for each activity during the process, and then give support to select the best option based on such project characteristics as effort, criticality, etc. Another useful reference can be found in [36]. Although this reference is not COTS oriented, it shows one method of process tailoring.

For R3: The evaluation criteria can be defined in a hierarchical manner using goal-oriented definition [37]. Some COTS selection approaches already use goals [31]. However, it is not clear yet how to estimate the satisfaction of each goal. For this purpose, the relationships between goals should be quantified in terms of how each goal contributes to or prevents the satisfaction of other goals. The work done in [38] can be used as a reference for this purpose.
For $\mathbb{R}_4$: There are many possible sources that explain the decision support concept [39-41]. An initial effort was done in [32] to incorporate decision support techniques in the COTS selection process.

For $\mathbb{R}_5$: Negotiation components are essential parts of decision support systems [6]. Group decision making techniques [43, 44] might be very useful here. Also, agent-based systems can be employed to support the same purpose [45].

For $\mathbb{R}_6$: The concept of learning software organization (LSO) [39] can be useful to help designing the required knowledge repositories. An initial effort was proposed in [46] where a framework for evaluating COTS with the support of knowledge bases is described. Also, there are useful online repositories for certain domains, e.g. [47]. However, more repositories are needed to cover the large spectrum of COTS domains.

For $\mathbb{R}_7$: The work proposed by the approach A6 [48] (see Appendix-1) shows how to adjust evaluation results based on the credibility of the data source. However, different vendors should have different credibility factors which could be estimated using techniques that can handle uncertain qualitative information, e.g. Bayesian Networks [49].

6. Conclusions

In this paper, we explored the evolution of COTS selection practices, and we surveyed and compared 18 of the most significant COTS selection approaches. In spite of the great variety of these approaches, there still many open issues related to the COTS selection process that need further research. We have listed some of these issues and showed some possible venues that could be taken to address them.

7. References


Appendix-I: Current COTS Selection Approaches in a Nutshell

This appendix summarizes current COTS selection approaches. The approaches are described in tables 2 to 19. For each approach, the following items are described:

- **Main Idea:** The main idea that distinguishes this approach from other approaches.
- **Main Steps:** The main steps of the approach.
- **Pros:** The evaluation strategy used by the approach.
- **Criticism:** The pros and cons of the approach.
- **TS:** The availability of tool support (TS).

Please refer to sections 2.1 and 2.2 for further details about these items.

Many of the approaches described in this appendix share the following cons:

- **REQ-ASSUMPTION:** The approach assumes the requirements already exist and fixed.
- **Multi-COTS:** The approach does not support multiple COTS selection for COTS intensive systems.
- **MISMATCHES:** It is not clear how to deal with COTS mismatches. The approach assumes that there is a set of COTS products that satisfy most of the requirements, at least to an acceptable level.
- **AHP/WSM-WEAKNESSES:** The approach uses AHP or WSM, and therefore inherits their weaknesses; e.g. consolidating the overall score is misleading, because strong aspects mask weak ones.

The acronyms described above are used when describing the approaches.

### Table 2 The OTSO Approach

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Idea</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td>1. Evaluation criteria: define evaluation criteria.</td>
</tr>
<tr>
<td>2. Search: search the market for possible COTS.</td>
</tr>
<tr>
<td>3. Screening: filter out the COTS that do not comply with the must-have requirements.</td>
</tr>
<tr>
<td>4. Evaluation: evaluate the benefit and cost of COTS.</td>
</tr>
<tr>
<td>5. Analysis: use AHP/WSM to consolidate the evaluation results and select a COTS.</td>
</tr>
<tr>
<td>6. Deployment: integrate the selected COTS into system.</td>
</tr>
<tr>
<td>7. Assessment: assess the success of the selection process as a feedback for future uses.</td>
</tr>
<tr>
<td><strong>Eval. Str.</strong></td>
</tr>
<tr>
<td><strong>Pros</strong></td>
</tr>
<tr>
<td><strong>Criticism</strong></td>
</tr>
<tr>
<td><strong>TS</strong></td>
</tr>
</tbody>
</table>

### Table 3 The IusWare Approach

<table>
<thead>
<tr>
<th>A2: IusWare Approach (1997) [17]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Idea</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td>1. Identify the actors relevant to the evaluation, their role, objective of the evaluation, and available resources.</td>
</tr>
</tbody>
</table>
Table 4 The PRISM Approach

<table>
<thead>
<tr>
<th>Main Idea</th>
<th>A3 : PRISM Approach (1997) [18]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The PRISM (Portable, Reusable, Integrated, Software Modules) approach comprises two parts: (i) a generic component architecture, and (ii) a product evaluation process (PEP). The PEP process focuses on prototyping to ensure the selected product complies with industry standards represented by the generic architecture.</td>
</tr>
<tr>
<td>Main Steps</td>
<td>1. Identification: based on initial criteria, identify products that fit into the generic architecture.</td>
</tr>
<tr>
<td></td>
<td>2. Screening: identify one or more products with best fitness for further examination.</td>
</tr>
<tr>
<td></td>
<td>3. Stand-alone test: evaluate products against reliability, reusability, and system requirements.</td>
</tr>
<tr>
<td></td>
<td>4. Integration test: estimate how readily the product is to be integrated into the architecture.</td>
</tr>
<tr>
<td></td>
<td>5. Field test, re-evaluate the product after deployment in the target context.</td>
</tr>
<tr>
<td>Eval. Str.</td>
<td>Progressive filtering</td>
</tr>
<tr>
<td>Pros</td>
<td>Address make-or-buy decisions.</td>
</tr>
<tr>
<td></td>
<td>Address issues related to the integration into system.</td>
</tr>
<tr>
<td></td>
<td>Emphasize the importance of prototyping: first outside the context, and second inside the context.</td>
</tr>
<tr>
<td>Criticism</td>
<td>Identifying a generic architecture can be in many cases valid only for specific context.</td>
</tr>
<tr>
<td></td>
<td>Many aspects of the approach are vague, e.g. how to acquire the requirements?, how to consolidate the results?, how to define the test cases?, etc.</td>
</tr>
<tr>
<td>TS</td>
<td>Not available.</td>
</tr>
</tbody>
</table>

Table 5 The CISD Approach

<table>
<thead>
<tr>
<th>A4 : CISD Approach (1997) [19]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Idea</td>
</tr>
<tr>
<td>Main Steps</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Eval. Str.</td>
</tr>
<tr>
<td>Pros</td>
</tr>
<tr>
<td>Criticism</td>
</tr>
<tr>
<td>TS</td>
</tr>
</tbody>
</table>

Table 6 The PORE Approach

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The PORE (Procurement-Oriented Requirements Engineering) approach focuses on the requirements engineering phase of the COTS procurement process. It suggests acquiring the requirements as the same time as acquiring and analyzing COTS products. PORE suggests iterating between requirements acquisition and product selection and rejection until finding a COTS that satisfies a sufficient number of the requirements (i.e. progressive filtering evaluation). PORE integrates techniques such as: feature analysis techniques [54] to help scoring the compliance of COTS to requirements, and multi-criteria decision making techniques [7] to help selecting a COTS.</td>
</tr>
<tr>
<td>Main Steps</td>
<td>1. Scoping evaluation effort: set the expected effort and schedule for the evaluation activities.</td>
</tr>
<tr>
<td></td>
<td>2. Searching / Screening: search for COTS candidates.</td>
</tr>
<tr>
<td>Eval. Str.</td>
<td>Progressive filtering</td>
</tr>
<tr>
<td>Pros</td>
<td>The parallel requirement acquisition and COTS selection means the defined requirements inform the selection process and vice versa, which is more realistic than assuming a fixed set of requirements.</td>
</tr>
<tr>
<td></td>
<td>PORE suggests using the Analytic Hierarchy Process (AHP) method [7] only after Phase3 after defining a small number of products. this leads to minimizing the extra effort that is required by the AHP method.</td>
</tr>
<tr>
<td>Criticism</td>
<td>PORE partially addresses the COTS mismatch problem. The progressive definition of the system requirements allow to refine those requirements in a way that reduces the mismatches with the products specifications. However, it is not clear it is till not clear how to deal with the situation when many mismatches are still unresolved. PORE does not define a systematic process that can be applied to (i) identify the influence of the remaining mismatches on the COTS selection decision, and (ii) to handle the remaining mismatches after selection.</td>
</tr>
<tr>
<td>TS</td>
<td>PORE Process Advisor (prototype tool)</td>
</tr>
</tbody>
</table>

Table 7 The CEP Approach

<table>
<thead>
<tr>
<th>A6 : CEP Approach (1999) [48]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Idea</td>
</tr>
<tr>
<td>Main Steps</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
The STACE Approach (1999) [14, 20, 55, 56]

**Main Idea**
STACE (Social-Technical Approach to COTS Evaluation) emphasizes the importance of non-technical issues when defining the evaluation criteria and conducting the evaluation process. The non-technical issues include social, human, and organizational characteristics, e.g., political and economic factors. STACE also emphasizes the customer participation during evaluation. In [55], studies were conducted to compare the application of STACE in developing and developed countries.

**Main Steps**
1. **Requirements elicitation**: the high level requirements are elicited from stakeholders, market studies, system documents, and domain knowledge.
2. **Social-technical criteria definition**: the high level requirements are decomposed into measurable attributes. The decomposition addresses the non-technical characteristics (e.g., social-economic).
3. **Alternatives identification**: the high level requirements are decomposed into measurable attributes. The decomposition addresses the non-technical characteristics (e.g., social-economic).
4. **Evaluation**: the COTS alternatives are evaluated and ranked according to the social-technical criteria.

**Eval. Str.**
- **Keystone evaluation**, and **Progressive filtering**

**Pros**
- Incorporate the non-technical characteristics in the evaluation process.
- Support the negotiation between requirements elicitation and COTS evaluation through emphasizing customer involvement.

**Criticism**
- STACE is a very extensive process that requires more effort to apply.
- Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES.

**TS**
- Not available.

---

The CRE Approach (1999) [57]

**Main Idea**
CRE (COTS-based Requirements Engineering) emphasizes the importance of non-functional requirement (NFR) as a decisive criteria when comparing COTS alternatives. CRE uses NFR framework [12] to model the NFRs for this purpose. Evaluating all NFRs requires more effort than most organizations have. So, CRE suggests select the most promising COTS candidates for detailed evaluation.

**Main Steps**
1. **Identification**: the core requirements and COTS candidates are identified.
2. **Description**: further requirements are defined. The NFR framework is used to model the NFRs.
3. **Evaluation**: COTS candidates are evaluated and based on 'must-have' criteria, and filter out the ones that do not satisfy them.
4. **Criteria definition**: give a detailed definition of the evaluation criteria. CEP defines several categories of evaluation criteria: functional, basic, architecture, management, and strategic.
5. **Evaluation**: evaluate COTS alternatives and estimate the CF factor for each criterion.
6. **Analysis**: analyze evaluation results and compare COTS alternatives.

**Eval. Str.**
- **Progressive filtering**
- **Using the NFR framework during COTS evaluation facilitates addressing the NFRs.**
- **Evaluating all NFRs (even for small set of COTS candidates) adds extra effort that is not available in most real situations.**
- **It is not clear how quality issues are verified, e.g. how a required level of quality is reached.**
- **It is not clear how to deal with unsatisfied quality attributes.**

**Criticism**
- Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES.

**TS**
- Not available.

---

The CARE Approach (1999) [57]

**Main Idea**
CARE (CARE-Aware Requirements Engineering) approach focuses on the requirements engineering phase of the selection process. CARE defines two types of requirements: native (requirements acquired from customers) and foreign (requirements that may be implemented by existing COTS products). CARE tries to bridge the gap between the native and the foreign requirements by either asking the customers to change a goal or requirements or asking the vendors to customize the COTS product.

**Main Steps**
1. **Define goals**: define system goals (i.e. native goals)
2. **Match goals**: search for COTS products that match the goals (functional first, non-functional second).
Table 13 The BAREMO Approach

<table>
<thead>
<tr>
<th>Main Idea</th>
<th>The BAREMO (Balanced Reuse Model) focuses on how to use the AHP technique [7] to select COTS products.</th>
</tr>
</thead>
</table>
| Main Steps| 1. Specify project objectives  
2. Construct a decision tree (where root node is the problem objective, intermediate nodes are criteria, leaf nodes are alternatives)  
3. Generate a comparison matrix for criteria at the same level in the decision tree  
4. Value each alternative at the leaf nodes with respect to connected criteria at intermediate nodes  
5. Calculate a final value for each alternative using weighted addition scales. |

<table>
<thead>
<tr>
<th>Eval. Str.</th>
<th>● Progressive filtering</th>
</tr>
</thead>
</table>
| Pros      | ● Suggest to use flexible requirements which is more realistic than defining a fixed set of requirements  
● Partially addresses the COTS mismatch problem  
● Support the definition of a plausible set of requirements based on different agents views. |
| Criticism | ● Although CARE emphasizes the importance to map system requirements to product specs, the following weaknesses was identified: CARE suggests negotiating the resolution of the mismatches and how this influences the system and the product. However, this was given as a high level guideline which means, there is no systematic process that can be applied to (i) identify the influence of these mismatches on the COTS selection decision, and (ii) to handle the identified mismatches. What CARE suggests is similar to what currently happens (in an ad-hoc manner) in real-world selection.  
● Not clear how to define requirements for multiple COTS selection in COTS intensive systems. |
| TS        | ● CARE Assistant Tool (prototype tool) |

Table 14 The Storyboard Approach

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Idea</strong></td>
</tr>
</tbody>
</table>
| **Main Steps** | 1. Determine the requirements that can be satisfied by COTS products.  
2. Develop use-cases for the identified requirements.  
3. Identify and evaluate COTS products based on Steps 1 and 2.  
4. Based on the use-cases in Step2, develop storyboard using screen captures. Screen captures can be taken from actual COTS products, or from a custom interface developed using e.g. JAVA. |
| **Eval. Str.** | ● Progressive filtering and Key-stoning. |
| **Pros** | ● Provide a meaningful way to communicate requirements among stakeholders more clearly and concisely.  
● Provide a means to manage users’ expectations.  
● Support the selection of multiple COTS components in COTS intensive systems. |
| **Criticism** | ● Non-functional requirements are not addressed.  
● No formal evaluation process is presented.  
● Not clear how to handle possible conflicts between the COTS products integrated together; i.e. resource usage.  
● MISMATCHES. |
| **TS** | ● Not available. |

Table 15 The Combined-Selection Approach

<table>
<thead>
<tr>
<th>A14 : Combined Selection Approach (2002) [29]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main Idea</strong></td>
</tr>
</tbody>
</table>
| **Main Steps** | 1. Plan the global selection process and firing individual selection processes. (global level)  
2. Identify COTS candidates for individual areas. (local level)  
3. Identify global COTS integration scenarios; e.g. product A will cover area 1, while product B will cover area 2 and part of area 1. (global level)  
4. Evaluate individual scenarios at each individual area. (local level)  
5. Evaluate integration scenarios (global level).  
6. Select the COTS products. |
| **Eval. Str.** | ● Depends on the process used at the local level; e.g. progressive filtering, if PORE is used. |
| **Pros** | ● Support the selection of multiple COTS components in COTS intensive systems.  
● CS inherits the weaknesses of the method used at the local level.  
● No formal models are presented to guide the decision making process.  
| **Criticism** | ● Not applicable.  
● Does not define a COTS evaluation / selection mechanism.  
● Although BAREMO investigates the use of AHP in details, the concept itself is not new and has been already used in many other approaches.  
● REO-ASSUMPTION, Multi-COTS, MISMATCHES, AHP/WSM-WEAKNESSES. |
### Table 16 The WinWin Approach

**Main Idea**
The WinWin spiral model is a risk-driven approach adapted from the classical software development spiral model [62]. WinWin follows an iterative process in which the requirements are acquired in parallel to evaluating the COTS products. WinWin emphasizes concurrent product identification and process refinement. A decision framework is used to provide guidance for the COTS-based development decisions, e.g., make-or-buy, COTS selection, COTS tailoring, glue-coding, etc.

**Main Steps**
- WinWin uses five iterative steps, in which more stakeholders and OC&P (Objectives, Constraints and Priorities) are identified in successive iterations, and more refinement of the process is applied:
  1. Identify stakeholders, and system OC&P
  2. Evaluate products with respect to OC&P, and address risks (e.g. related to customer support)
  3. Elaborate product and process definition
  4. Verify and validate product and process definition.
  5. Stakeholders’ review and commitments

**Eval. Str.**
- Progressive filtering

**Pros**
- Addressing risks early in the selection process reduces the cost of handling them in the future.
- The decision framework provides an inclusive description of the actual development process.
- Gradually defining the stakeholders’ needs as well as the process fits more in many real-world situations.
- WinWin addresses the selection of multiple COTS components in intensive COTS-based systems.
- Although multiple COTS selection is support, no detailed guidelines (or formal process) were defined to elaborate this issue.
- Although WinWin emphasizes the importance to address risks (including those related to COTS mismatches) during its spiral process, this was given as a set of high level guidelines. That is, there is no systematic process that can be applied to (i) identify the influence of these mismatches on the COTS selection decision, and (ii) handle the identified mismatches after selection.

**Criticism**
- Not available.

**TS**
- Not available.

### Table 17 The Approach by Erol et al.

**Main Idea**
The approach by Erol et al. is an evaluation approach that supports selecting a COTS product from a finite set of products based on: (i) more than one objective and (ii) a set of quantitative (e.g. cost) and qualitative (e.g. linguistic variables) data. The approach uses fuzzy QFD (Quality Function Deployment) [63] to collect and quantify the qualitative data. Then, goal programming (a generalization of linear programming) is used to suggest near-optimal solutions to the decision maker.

**Main Steps**
- Acquire requirements and product information.
- Transform qualitative data to quantitative data using fuzzy QFD.
- Construct a multi-objective model for objectives: (i) maximize customer value, and (ii) minimize cost.
- Solve the model several times with slightly different parameter values to get multiple feasible solutions from which the decision maker can choose one solution.

**Eval. Str.**
- Not applicable

**Pros**
- Address qualitative data.
- Provide more than one feasible solutions to support making the final decision.
- Do not address activities other than COTS evaluation.

**Criticism**
- REQ-ASSUMPTION, Multi-COTS, MISMATCHES.

**TS**
- Not available.

### Table 18 The DesCOTS system

**Main Idea**
DesCOTS (Description, evaluation and selection of COTS components) system includes a set of tools that can be used to evaluate COTS products based on quality models such as ISO/IEC9126 [13]. DesCOTS is built on the work proposed in [11]

**Main Steps**
- DesCOTS follows the GCS process and uses the following steps to define the evaluation criteria:
  1. From ISO/IEC9126, determine the quality characteristics and subcharacteristics that will be used.
  2. Refine the subcharacteristics as needed, e.g. ‘suitability’ to basic suitability, and added suitability.
  3. Refine subcharacteristics into measurable attributes.
  4. Refine description of the actual development process.
  5. Identify relationships between quality entities.

**Eval. Str.**
- Progressive filtering

**Pros**
- Detailed method to define the evaluation criteria.
- Considers quality characteristics.
- Can be adapted to many domains.

**Criticism**
- Not available.

**TS**
- Yes

### Table 19 The MiHOS Approach

**Main Idea**
MiHOS (Mismatch-Handling aware COTS Selection) focuses on handling the mismatches between COTS candidates and the requirements. MiHOS firstly provides guidelines to handle the mismatches that are detected during the COTS selection process, and secondly uses decision support techniques to (i) study the cost of resolving these mismatches on the COTS selection decision, and (ii) support handling the mismatches of the selected COTS product.

**Main Steps**
- MiHOS has two components: the “COTS selection” component which follow exactly the GCS process, and the “mismatch handling” component which is divided into two parts:
  - Part1 (of mismatch handling component):
    1.1 Analyze the detected mismatch
    1.2 Decide whether the mismatch should be tolerated or resolved by (i) adjusting the requirements, or by (ii) adjusting the COTS product. If (ii) is chosen, then the mismatch is postponed until a COTS is selected.
  - Part2 (of the mismatch handling component):
    2.1 Modeling: define problem parameters, e.g. available resources for resolving the mismatches of the selected COTS product, the possible techniques to resolve each mismatch, cost of each technique, etc.
    2.2 Exploration: use optimization techniques to find optimal or near optimal solutions.

**Eval. Str.**
- Keystone identification / Progressive filtering

**Pros**
- Can handle mismatches during/after selection process.
- Uses goal-driven criteria definition which allows for more informative decisions.
- Use interactive decision support techniques which enable decision makers to have more participation in and control over their decisions.

**Criticism**
- Requires effort more than most selection approaches.
- Is not intended to address quality requirements.
- No support for the selection of multiple COTS components in COTS intensive systems.

**TS**
- MiHOS-TS, a prototype.