STACE

(Social Technical Approach to COTS Software Evaluation)

Draft paper

An introduction to STACE

“STACE” stands for “Social Technical Approach to Commercial-of-the-shelf (COTS) Software Evaluation. It is a framework for selecting COTS software. “[...] STACE was developed to facilitate a systematic requirements-driven COTS software selection and this problem using social-technical techniques.” (Kunda, 2003). The STACE framework tries to address not only the technical side of the COTS software selection process, as many other selection techniques do as well, but also the social side. STACE tries to combine the “soft” issues as the cost, the vendor reputation and capabilities and the organizational issues as well as the technical issues of the COTS software selection process.

As underlying decision method STACE uses the Analytical Hierarchical Process (AHP) (Saaty, 1980), a method designed for making complex multi-attribute decisions. STACE makes sure all the relevant factors and areas of interest for successful COTS software
selection are covered in the criteria on which the AHP rates the different alternatives. Neither STACE, nor AHP provide specific roles to be assigned.

**Steps and deliverables**
The initial description of STACE contains four interrelated processes. However, when Kunda (2003) continues to the case study of using STACE to select COTS Software, he recognizes six different steps. As the steps cover all the different activities in STACE better than the four processes, the six steps will be used for the further elaboration on STACE. At each step the corresponding process will be stated (if applicable).

**Step 1: Requirements definition (Process 1)**
In the first step, all the stakeholders are identified, as are the high level user requirements. These stakeholders come together in Joint Application Development (JAD) sessions and review meetings. In these meetings, they define the lower level requirements for the to be selected COTS software. STACE recommends the use of JAD sessions and review meetings for defining requirements, because traditional requirement selection methods tend to focus more on the technical issues, neglecting the social issues.

The high level requirements also contain the keystone characteristics. These are the characteristics a COTS component must have in order to be eligible. Keystone characteristics are often of a technical nature. For example, if a company runs a Microsoft Windows based network environment, the software to be selected must support Windows clients.

*Deliverables of step 1*
- Deliverable 1: List of stakeholders
- Deliverable 2: High level user requirements
- Deliverable 3: Keystone characteristics
- Deliverable 4: Low level user requirements

**Step 2: Social-technical criteria definition (Process 2)**
When all the requirements are defined, they are decomposed into four different social-technical criteria. 1) Functionality characteristics, 2) technology factors, 3) product quality characteristics and 4) social-economic factors. Within these four categories, requirements are placed in a hierarchy, on which in the next step, their relative importance will be determined. An other characteristic of an Alternative are the cots. This is not a part of STACE but should of course not be forgotten to take into account.

In this step keystone characteristics are not divided into the different criteria. As the software must be compliant with the keystone, a keystone is a filter, not a characteristic on which the COTS-components can be rated.

*Deliverables of step 2*
- Deliverable 5: Hierarchy of criteria, the “Social-technical criteria for COTS software selection”. For easy generation of this STACE deliverable, a template is created. For this template please refer to Appendix A.

**Step 3: Determine relative importance of the criteria using AHP.**
In this step, the relative importance of the criteria is calculated using pairwise comparison. This pairwise comparison is done per branch of the hierarchy, per level. See also table 2 and table 3 in (Kunda, 2003).
Deliverables of step 3
Deliverable 6: Overview of the relative importance of the criteria. (if specialized software is used to calculate these relative importances, it could be impossible to automatically generate such an overview)

Step 4: Identify candidate software (vendors) (Process 3)
In the alternatives identification process COTS components are selected that meet the high level requirements. These components can then be evaluated more in-depth. STACE recommends the use of several different techniques to identify candidate COTS products, including networking, internet search, market surveys, invitation to tenter (ITT), or request for proposals (RFP), vendor promotions and publications. In some cases it is possible to use evaluation copies of the software to collect even more information on the COTS product.

Deliverables of step 4
Deliverable 7a: List of alternatives
Deliverable 7b: In depth information on the alternatives. (this information is needed for the pairwise comparison per criteria, done in step 5)

Step 5: Evaluation (assessment) (Process 4)
In this step the actual evaluation of all the alternatives takes place. This is done on the basis of the information collected in Step 4.

Using specialized software (calculating all the pair comparisons by hand is only recommended by small samples) (Noszczyński et al, 2009) for each pair of software, a comparison is made on each criteria.

Deliverables of step 5
Deliverable 8: Overview of the priority ranking of the alternatives, per criteria. (as with the “overview of relative importance of the criteria” in step 3, it might be possible such an overview is not available, or only available per criteria)

Step 6: Using AHP to synthesize the evaluation results and select the product
In this final stage of STACE, the outcomes of step 3 and step 5 are combined to calculate the final scores of the alternatives. This is, in short (see (Saaty, 1980) for more information on how AHP works), done by combining the relative importance of the criteria, with the relative score of the alternatives per criteria. At the end of step 6, all the alternatives have a relative score on each of the four categories (See step 2). By combining these relative scores with the relative score per category, a final score per alternative is calculated.

Deliverables of step 6
Deliverable 9: Table with the relative score per alternative per category and the final relative score per alternative. This table indicates which COTS component is best suitable, given the entered pairwise comparisons.

Example of using STACE
In the following example the usage of STACE will be shown. For the AHP part of the example (Noszczyński et al, 2009) will be used.
Introduction

“Sellalot” is a company selling business gifts. The company sells everything on which a company logo can be printed. From umbrellas to ballpoint and from t-shirts to staplers. They are planning to open a new store in the hart of Utrecht. To be able to run this store smoothly, a few computers are installed. These computers, running Windows XP™ are connected to their (Microsoft Windows based) domain and are equipped with the custom build ERP system Sellalot uses. At the moment in all the other Sellalot offices, the Microsoft Office 2003 suite is used. Since the MS Office 2003 suite is becoming out-dated, Sellalot does not want to invest in new 2003 licenses for the new Utrecht office. They turned to their usual IT consultant for help with the selection of a new (COTS) software package. This package must be compatible with the documents currently created in the other offices of Sellalot. Since the custom built system Sellalot uses contains a mail client and takes care of the financial administration, the most important thing for Sellalot is the word-processor.

Using STACE

In this section we will walk through the six steps of STACE to select the best office product for the new office of Sellalot. For conducting all the necessary pairwise comparisons, the software of Noszczyński et al (2009) will be used. This software, called the “AHPproject”, is a free web based tool for conducting pairwise comparison using the AHP method. This is because calculating over 40 pairwise comparisons and their relative importance can be both very tedious and error sensitive.

Step 1: Requirements definition

After the stakeholder identification (deliverable 1), the stakeholders come together in a meeting to define the requirements for the new office suite. In this case, first the keystone criteria (deliverable 3) are laid-out by the IT department and the Management. Then the meeting continues to specify the higher (deliverable 2) and lower level (deliverable 4) requirements. It can also be possible the keystone criteria are selected after the determination of the High level user requirements.

Deliverable 1 - List of Stakeholders
- Management
- Employees
- IT department

Deliverable 2 - High level requirements
- Read other popular file formats
- Extensive help files
- Template support
- Supports the Open Document Format (ODF)
- Mail merge support
- Stable
- Reputation of vendor

Deliverable 3 - Keystone criteria
- Able to run on the Windows XP OS
- Compatible with Office 2003 documents
**Deliverable 4 - Lower level requirements**
- PDF export function
- Track changes function
- Simultaneous editing of a document
- Support in Dutch
- Cost
- Looks like Office 2003

**Step 2: Social-technical criteria definition**
When all the requirements are defined, they are decomposed into the four different social-technical criteria (deliverable 5).

**Deliverable 5 - Hierarchy of criteria**
Figure 1 gives a good impression on how such a hierarchy would look like. Note that in this step, the keystone characteristics are not divided into the different categories. As compatibility with the Windows platform was the only technical requirement, this category is left out in the Hierarchy. Cost is added because cost is always an issue.

![Hierarchy of criteria (deliverable 5).](image)

**Step 3: Determine relative importance of the criteria using AHP.**
Next the relative important of the criteria is determined. The software used (Noszczyński et al, 2009) is not capable of showing a table, so some screenshots are made for illustrational purposes figure 2 and figure 3.
Deliverable 6a - Relative importance of criteria (highest level in hierarchy)

![Relative importance of criteria - highest in hierarchy (deliverable 6a).]

Deliverable 6b - Relative importance of criteria (lower level in hierarchy)

![Relative importance of criteria - lower in hierarchy (deliverable 6a).]

Step 4: Identify candidate software (vendors)

After the requirement phase, Sellalot searches for different office suites. After the keystone criteria are taken into account, the following three office suites remain.
Deliverable 7 - List of alternatives
- Microsoft Office 2007
- OpenOffice.org 3.0
- Word Perfect Office X4

Step 5: Evaluation (assessment)
Fortunately Sellalot was able to gain access to all three office suites, so they could be thoroughly evaluated and tested. Using the AHP project software, for each pair of software, a comparison is made on each criteria.

Deliverable 8 - Priority ranking per alternative, per criteria
The AHP project software used does not allow you to output a table of all the rankings per alternative per criteria. Therefore an overview of only one criteria is shown, the “PDF output feature” figure 4.

![Figure 4. Priority ranking of the PDF output feature (deliverable 8).](image)

Step 6: Using AHP to synthesize the evaluation results and select the product
In this final stage of STACE, the outcomes of step 3 and step 5 are combined to calculate the final scores of the alternatives. This is, in short (see Saaty (1980) for more information on how AHP works), done by combining the relative importance of the criteria, with the relative score of the alternatives per criteria. At the end of step 6, all the alternatives have a relative score on each of the four categories (See step 2). By combining these relative scores with the relative score per category, a final score per alternative is calculated (deliverable 9) figure 5.
Deliverable 9 - Final score per alternative.

**Figure 5. Final score of the three alternatives (deliverable 9).**

**STACE Meta-Models**

Using the techniques proposed by Weerd et al (2008), STACE was analyzed and the following meta models were constructed.

In this assignment the STACE method is described using the techniques described by Weerd et al (2008). First all the processes are modeled in the meta process model. After that the concepts in use are modeled in the meta deliverable model. Finally these two models are combined in the process deliverable diagram (PDD).

Both the activities and the concepts are separately described in table 1 and table 2.

**Meta-process model**

The meta-process model consists of all the processes mentioned in the description of the method. Although the description of STACE by Kunda (2003) does not specify any particular roles, it is possible to assign them to the different activities. Two roles can be distinguished in STACE 1) the (end) user and 2) the consultant. The necessity of these two roles is in line with two of the principals behind STACE.

- “Support for evaluation of both COTS products and the underlying technology”, this principal indicates that domain knowledge on the subject of COTS products and technology is needed for successful evaluation.
- “Use of social-technical techniques to improve the COTS selection process”, this principal indicates that “User participation is regarded as an effective strategy [...]” and successful selection of a COTS product cannot take place without an user.
The consultant is there to support the user in the process of selecting a suitable COTS product, he is not only the one with knowledge of the potential alternatives, but also the one with knowledge about STACE, the AHP and the software used to implement the AHP. Therefore the consultant plays a role in every step of STACE. The user on the other hand, plays a role in only four of the steps. This is because the user is not needed in the selection of suitable ALTERNATIVES, as the user has not enough domain knowledge to perform this activity. Also the final calculations using the AHP (software) is a task for someone with knowledge on that particular subject, the consultant.

As mentioned before, the STACE method consist of six steps, 3 standard activities and 3 complex activities (gray), which can be identified in figure 6 and are all taken from Kunda (2003).
Figure 6. The meta process model of STACE. (KUNDA, 2003)
### Activity Table

The activity table contains all the activities from the meta process model. When an activity does not have any sub activities, the cells of the table are merged.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Sub activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement definition</td>
<td>Identify Stakeholders</td>
<td>A list of STAKEHOLDERS is made, so the JAD* sessions in which the REQUIREMENTS will be determined can be organized.</td>
</tr>
<tr>
<td>Determine high level user requirements</td>
<td></td>
<td>The HIGH LEVEL USER REQUIREMENTS are determined</td>
</tr>
<tr>
<td>Determine lower level user requirements</td>
<td></td>
<td>The LOWER LEVEL USER REQUIREMENTS are determined</td>
</tr>
<tr>
<td>Determine Keystone characteristics</td>
<td></td>
<td>The KEYSTONE CHARACTERISTICS are selected from the HIGH LEVEL USER REQUIREMENTS</td>
</tr>
<tr>
<td>Social-Technical criteria definition</td>
<td>Decompose Criteria</td>
<td>The CRITERIA are decomposed into the four categories that STACE* acknowledges</td>
</tr>
<tr>
<td></td>
<td>Place requirements in hierarchy</td>
<td>The REQUIREMENTS are placed in a HIERARCHY which will suite their relative importance best and shows which REQUIREMENTS are interrelated.</td>
</tr>
<tr>
<td>Relative importance calculation of the criteria using AHP</td>
<td></td>
<td>Using pairwise comparison, of all the criteria per node per level, their relative importance is calculated. This is done using AHP*, preferably with the use of specialized software.</td>
</tr>
<tr>
<td>Candidate software (vendor) identification</td>
<td>Identify candidate software (vendors)</td>
<td>A list of ALTERNATIVES or candidate software (vendors) is created</td>
</tr>
<tr>
<td></td>
<td>Gather in-depth information on alternatives</td>
<td>Extensive information is gathered on the ALTERNATIVES, covering all the CRITERIA on which the ALTERNATIVES will be rated on in the next step.</td>
</tr>
<tr>
<td>Evaluate alternatives per criteria per pair</td>
<td></td>
<td>Pairwise comparison is done to evaluate every combination of ALTERNATIVES per CRITERIA.</td>
</tr>
<tr>
<td>Synthesize the evaluation results</td>
<td></td>
<td>The outcomes of “Calculate relative importance of the criteria using AHP” and “Evaluate alternatives per criteria per pair” are combined to calculate the overall relative score of each ALTERNATIVE and come up with the best ALTERNATIVE.</td>
</tr>
</tbody>
</table>

*Although in capital, not meant to be concepts.

Table 1. STACE Activity Table. (Kunda, 2003)

### Meta-deliverable model

The meta-deliverable model consists of all the concepts and deliverables the STACE method contains. The first thing that needs to be stated is that the STAKEHOLDER LIST is a concept that is only necessary for the successful usage of STACE. The STAKEHOLDER
LIST contains all the stakeholder of the to be selected COTS software system and is used to gather attendees for the JAD session in which the USER REQUIREMENTS are defined.

The USER REQUIREMENTS are divided into two categories, HIGH LEVEL USER REQUIREMENTS and LOW LEVEL USER REQUIREMENTS. A HIGH LEVEL USER REQUIREMENT can be a KEYSTONE CHARACTERISTIC. This means that this criteria must be met by an ALTERNATIVE to be suitable for comparison. All the USER REQUIREMENTS are placed in one of the four categories of STACE (Functionality, Technology, Product quality and Social-economic factors). After that, this subdivision forms the HIERARCHY OF REQUIREMENTS, together with pairwise comparisons of all the criteria using AHP.

After the pairwise comparisons, the consultant, a role not officially in STACE, but essential for a good outcome of the selection process, is searching the market for suitable ALTERNATIVES. At first he only pays attention to the KEYSTONE CHARACTERISTICS, but later on, when the list of ALTERNATIVES is completed, he gathers extensive information on all the ALTERNATIVES, in the meta-deliverable model shown as the attributes of the ALTERNATIVE concept.

At the end the PRIORITY RANKING OF ALTERNATIVES is composed using AHP out of the RELATIVE CRITERIA IMPORTANCE and the set of ALTERNATIVES. The PRIORITY RANKING OF ALTERNATIVES contains a final value for each of the ALTERNATIVES, these values are the input for the FINAL ALTERNATIVE RANKING. The full meta-deliverable model is showed in figure 7, and is based on Kunda (2003).
Figure 7. The meta-deliverable model of STACE (KUNDA, 2003)
Table of concept definitions

The table of concept definitions contains all the concepts from the meta-deliverable model and their description. All the concepts are written in CAPITAL.

Table 2. STACE concept Table. (Kunda, 2003)

<table>
<thead>
<tr>
<th>CONCEPT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STAKEHOLDER LIST</td>
<td>Contains a list of stakeholders. This document is necessary to make sure all the people who need to be at the JAD session can be invited.</td>
</tr>
<tr>
<td>USER REQUIREMENT</td>
<td>A criteria to which an ALTERNATIVE ought to comply.</td>
</tr>
<tr>
<td>CRITERIA</td>
<td>An other word for USER REQUIREMENT</td>
</tr>
<tr>
<td>HIGH LEVEL USER REQUIREMENT</td>
<td>A USER REQUIREMENT of which it is very important that an alternative complies with it.</td>
</tr>
<tr>
<td>LOW LEVEL USER REQUIREMENT</td>
<td>A USER REQUIREMENT of which it is less important that an alternative complies with it.</td>
</tr>
<tr>
<td>KEYSTONE CHARACTERISTIC</td>
<td>A USER REQUIREMENT of which it is obligated an alternative complies with it.</td>
</tr>
<tr>
<td>CRITERIA DECOMPOSITION</td>
<td>Decomposition of all the criteria, based on their categories. STACE* recommends four different categories.</td>
</tr>
<tr>
<td>ALTERNATIVE</td>
<td>COTS* software component which might be suitable to fulfill the task to be done, must at least comply with the KEYSTONE CHARACTERISTICs</td>
</tr>
<tr>
<td>HIERARCHY OF REQUIREMENTS</td>
<td>Division of the USER REQUIREMENTS so their relative dependance and coherence can be determined.</td>
</tr>
<tr>
<td>RELATIVE CRITERIA IMPORTANCE</td>
<td>Through pairwise comparison gained relative importance of the USER REQUIREMENTS on which the ALTERNATIVEs will be compared</td>
</tr>
<tr>
<td>PRIORITY RANKING OF ALTERNATIVES</td>
<td>Total set of pairwise comparisons per USER REQUIREMENT pair of the HIERARCHY OF REQUIREMENTS.</td>
</tr>
<tr>
<td>FINAL ALTERNATIVE RANKING</td>
<td>Final calculation based on the PRIORITY RANKING OF ALTERNATIVES which contains the relative appropriateness per ALTERNATIVE</td>
</tr>
</tbody>
</table>

*Although in capital, not meant to be concepts.

Process Deliverable Diagram

This Process Deliverable Diagram (PDD) contains all the activities and concepts from the meta process model and the meta deliverable model. The dotted lines show which activity delivers which concept. Figure 8 gives a complete overview of the STACE method.
Figure 8. Process Deliverable Diagram. (Kunda, 2003).
The Origin of COTS and STACE

AS the designing and building of software became more and more complex, the demand for already tested software that could do the job as well grew larger and larger (Chung et al, 2004). The advantages of COTS software are evident: It is already there, so you do not have to build it yourself, the vendor has (probably) tested it and the costs are often lower than when a custom system is built. Unfortunately, using COTS software also has its downsides. COTS software is never precisely what is needed. it (almost always) contains features that will be barely used, or lack features that could come in handy (Basili et al, 2001).

But how to select the right COTS software of component out of the enormous number of packages being offered? Since the mid-90s, more than a dozen methods have been developed. Land et al (2008) list about 17 different method versions, developed in the last 15 years and (adhav et al (2008) even state 27. For a short overview of the methods Land et al (2008) found, see table 3.

STACE (Kunda et al, 1999) was developed around 1999, because it was thought that existing methods at the moment payed too little attention to the non-technical issues considering COTS evaluation. STACE was influenced by the OTSO (Kontio et al, 1996) and the PORE (Ncube et al, 1999) method and based on the following principles (Kunda et al, 1999).

- Usage of appropriate techniques to evaluate the alternatives in a systematic way, instead of the ad-hoc manner in which they often are. Save lessons learned so future decisions can benefit from previous ones. Saving lessons learned and working more systematic will shorten evaluation time and lower the costs involved.
- Evaluate not only the COTS components which could be used, but also the techniques on which they are based.
- Make the end users part of the software selection process. Doing this will make sure the software will be better used and better accepted by its end users.

Table 3: different COTS selection methods. Taken from Land et al (2008)

<table>
<thead>
<tr>
<th>Year</th>
<th>Method</th>
<th>Main Novelty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>OTSO (Off-The-Shelf Option)</td>
<td>Progressive filtering; evaluation criteria includes functionality, non-functional properties, strategic considerations and architecture compatibility; AHP suggested for comparison</td>
</tr>
<tr>
<td>1997</td>
<td>PRISM (Portable, Reusable, Integrated, Software Modules)</td>
<td>Stand-alone test phase followed by integration evaluation and field test</td>
</tr>
<tr>
<td>1998</td>
<td>PORE (Procurement-Oriented Requirements Engineering)</td>
<td>Closely intertwined selection of components and definition of system requirements</td>
</tr>
<tr>
<td>Year</td>
<td>Method</td>
<td>Main Novelty</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1999</td>
<td>STACE (Socio-Technical Approach to COTS Evaluation)</td>
<td>Stresses importance of non-technical factors to evaluate</td>
</tr>
<tr>
<td>2000</td>
<td>COTS Score</td>
<td>-</td>
</tr>
<tr>
<td>2001</td>
<td>RCPEP (Requirements-driven COTS Product Evaluation Process)</td>
<td>Stresses evaluation objectivity</td>
</tr>
<tr>
<td>CAP</td>
<td>Large number of quality metrics (&gt;100)</td>
<td></td>
</tr>
<tr>
<td>i-MATE</td>
<td>Reusable requirements for middleware selection</td>
<td></td>
</tr>
<tr>
<td>PECA</td>
<td>Flexible structure of activities</td>
<td></td>
</tr>
<tr>
<td>RDR (Requirements and Design Reviews)</td>
<td>Explicitly describes the relation between acquired components and system parts being built in-house</td>
<td></td>
</tr>
<tr>
<td>CRE (COTS-Based Requirements Engineering)</td>
<td>Requirements engineering process drives the selection; NFR framework is used to discuss non-functional attributes</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>CSCC (Combined Selection of COTS Components)</td>
<td>Considers the total cost for a system rather than specifying in advance the individual costs for different components</td>
</tr>
<tr>
<td>2003</td>
<td>CEP (Comparative Evaluation Process)</td>
<td>-</td>
</tr>
<tr>
<td>2004</td>
<td>CARE (COTS-Aware Requirements Engineering)</td>
<td>Intertwines system requirements engineering with component evaluation; later named CARE/SA when giving software architecture a stronger focus</td>
</tr>
<tr>
<td>CCCS (Compatible COTS Component Selection)</td>
<td>Considers sets of complementary component as candidates, focusing on how well components will fit together; also emphasizes prototyping as a means to collect reliable information.</td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>CPF (Commitment, Pre-filtering, Final filtering)</td>
<td>Strong focus on continuous improvement of the selection process itself</td>
</tr>
<tr>
<td>2006</td>
<td>CSSP (COTS Software Selection Process)</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: different COTS selection methods. Taken from Land et al (2008) (continued).
Other ways to evaluate COTS software

As written before, there are over a dozen of COTS software evaluation techniques. After an extensive literature research in the COTS software evaluation field by Jadhav and Sonar (Jadhav et al, 2008), it was discovered that there were four techniques used in these methods. Below these techniques will be described with their strengths and weaknesses.

The Analytical Hierarchical Process. (AHP)

The AHP was developed by Saaty (1980) and is a systematic approach for multi-attribute decisions which has been widely accepted. It is capable of supporting individual as well as group decisions and can deal with both qualitative as quantitative criteria. It helps the decision-maker by structuring the criteria and ranking them according to their relative importance (Mustajoki et al, 2000).

Weaknesses of the AHP lay in the large amount of calculations that is needed to be done. Per alternative pair, per criteria a judgement has to be made on a scale from 1 to 9 (1 meaning “equal importance”, 9 “extreme importance”). If an alternative changes, or is deleted, relative scores have to be re-calculated. Relative scores can change completely when criteria are added or deleted. Of course a lot of these weaknesses can be overcome by using software to support the AHP (Noszczyński, 2009).

Feature Analysis (FA) (Kunda, 2003)

Feature analysis is a flexible way of evaluation, it is suitable for analyzing technical aspects as well as vendor features. Feature analysis is usable to analyze different sources of information, like case studies or formal experiments.

Weakness of the feature analysis method is that is can be difficult to produce an useful aggregate score per alternative, as different combinations of awarded points can produce the same score.

Weighted Sum/Score Method (WSM) (Ncube et al, 2002).

The Weighted Sum Method is often used because the method is easy to apply and can be calculated by hand or with a simple spreadsheet. However, the WSM outputs numbers, which can easily be mistaken for numbers with contain meaningful differences. Also, selecting criteria to fully cover all the aspects while minimizing overlap can as well be very hard.

Fuzzy based approach

The Fuzzy base approach is used when there is a lot of uncertainty. In such a case it is hard to assign precise weights and values to the criteria. The fuzzy approach makes it possible to use linguistic labels in stead of precise numbers. It tries to level with the way humans make decisions. The computations which are involved in the fuzzy approach are how ever quite complicated (Jadhav et al, 2008)(Lin et al, 2007).

Calculations

All of the above mentioned techniques use some kind of calculation to find the best COTS component. Some computations are quite straightforward (FA), while other methods (AHP) require complex calculations to be done. In different articles (Kunda, 2003);(Land et al, 2008);(Jadhav et al, 2008) and (Ncube et al, 2002), the necessity of complex calculations
is seen as a disadvantage. These calculations do take a considerable amount of time. However, for AHP for example, there are different software tools around to help you with the decision process. These tools make life a lot easier and even allow you to delete alternatives or criteria after the relative weights have been assigned (Noszczyński, 2009).

**Different applications of STACE**

In the last few years, not many scientists have done case studies with the STACE method. STACE is mentioned a lot, but always in papers in which different COTS evaluation methods are compared like (Land et al, 2008) and (Jadhav et al, 2008) or where new methods were constructed (Labed Jilani et al, 2005)

The only case study I found was done by the developer of the STACE method (Kunda, 2003). He tested the STACE method on the selection of a Geographic Information System (GIS) at “a public organization mandated to protect the environment and control pollution”. The outcome of this case study was, not surprisingly, positive. He pointed out that STACE succeeded in what is was initially designed for: “To address the technical as well as the non-technical criteria of COTS software.” The downside of the STACE method was that there were over 100 pairwise comparisons to make, this was a lot of “boring” work. However, the result of the pairwise comparisons was an AHP decision with an audit trail, which made the decision process transparent. Another advantage of the STACE method was that “[..] stakeholder participation as advocated in the STACE framework is very important in COTS software evaluation and selection as it facilitates dialogue and consensus building with stakeholders.” However, it was also mentioned that the stakeholder participation increased the cost of the evaluation process.

**References**


**Appendixes**

**Appendix A - A template for STACE deliverable 5.**

As this template is made to function in Microsoft Word 2007, it can be downloaded in the Microsoft Word format from the Method Engineering wiki: [http://www.cs.uu.nl/wiki/bin/view/MethodEngineering/STACE:AnApproachForCOTSComponentSelection#topic-actions](http://www.cs.uu.nl/wiki/bin/view/MethodEngineering/STACE:AnApproachForCOTSComponentSelection#topic-actions)

As Microsoft Word 2007 offers some visualization options not compatible with other word processors, only the textual part of the template will be published in this paper (page 21). The automatic generation of a graphical representation of the hierarchy will only work in Microsoft Word 2007.
Hierarchy of Criteria

Project sponsor: "Insert project sponsor"
Project name: "Insert project name"
Author: "Insert name" ("consultant/end user")
Date: Monday, 23 March, 2009
Version: "Insert version number"

Explanation:
- Fill in the requirements in the bold categories below.
- If it suits the selection criteria better, feel free to add extra sub-categories.
  (as shown in "functional requirement 2 (group)").
- After the hierarchy is finished, copy/paste it into the text field of the chart below,
  to generate an easy readable overview.
  (Unfortunately this only works in Microsoft Word 2007)

- Criteria
  - Functionality Requirements
    - "functional requirement 1"
    - "functional requirement 2 (group)"
      - "functional requirement 2.1"
      - "functional requirement 2.2"
      - "functional requirement 2.3"
    - "functional requirement 3"
    - "functional requirement 4"
    - ...
    - "functional requirement n"
  - Quality Attributes
    - "quality attribute 1"
    - "quality attribute 2"
    - "quality attribute 3"
    - "quality attribute 4"
    - ...
    - "quality attribute n"
  - Technical Characteristics
    - "technical characteristic 1"
    - "technical characteristic 2"
    - "technical characteristic 3"
    - "technical characteristic 4"
    - ...
    - "technical characteristic n"
  - Non-Technical Factors
    - "non-technical factor 1"
    - "non-technical factor 2"
    - "non-technical factor 3"
    - "non-technical factor 4"
    - ...
    - "non-technical factor n"
  - COST issues