Evaluation methods for procurement of business critical software systems

Nils Rosén
Evaluation methods for procurement of business critical software systems

Submitted by Nils Rosén to the University of Skövde as a dissertation towards the degree of B.Sc. by examination and dissertation in the School of Humanities and Informatics.

2009-06-07

I hereby certify that all material in this dissertation which is not my own work has been identified and that no work is included for which a degree has already been conferred on me.

Signature: _______________________________
Evaluation methods for procurement of business critical software systems
Nils Rosén

Abstract

The purpose of this thesis is to explore what software evaluation methods are currently available that can assist organizations and companies in procuring a software solution for some particular task or purpose for a specific type of business. The thesis is based on a real-world scenario where a company, Volvo Technology Corporation (VTEC), is in the process of selecting a new intellectual property management system for their patent department. For them to make an informed decision as to which system to choose, an evaluation of market alternatives needs to be done. First, a set of software evaluation methods and techniques are chosen for further evaluation. An organizational study, by means of interviews where questions are based on the ISO 9126-1 Software quality model, is then conducted, eliciting user opinions about the current system and what improvements a future system should have. The candidate methods are then evaluated based on the results from the organizational study and other pertinent factors in order to reach a conclusion as to which method is best suited for this selection problem. The Analytical Hierarchy Process (AHP) is deemed the best choice.

Key words: information systems, business critical software, evaluation methods, intellectual property management system, IPMS, AHP, MCDA, GQM, PECA
Table of contents

1 Introduction ........................................................................................................................................... 1

2 Background ........................................................................................................................................ 2

2.1 Information systems and organizational levels.............................................................................. 2

2.2 Document and information handling systems .............................................................................. 3

2.3 Application software & IPMS ........................................................................................................ 3

2.4 Business critical systems ............................................................................................................. 4

2.5 Failures of IS.................................................................................................................................. 4

2.6 IS investment evaluation................................................................................................................. 4

2.7 Evaluating an IS............................................................................................................................. 5

2.8 Evaluating quality of an IS ........................................................................................................... 6

2.9 The ISO Software Quality Model.................................................................................................. 6

2.10 Commercial-off-the-shelf (COTS) evaluation ............................................................................... 9

3 Problem statement ............................................................................................................................ 11

3.1 Presenting the problem.................................................................................................................. 11

3.2 Why the problem should be studied ............................................................................................. 11

3.2.1 The need for evaluation............................................................................................................. 11

3.2.2 Difficulty in selecting method .................................................................................................. 12

3.2.3 The problem from an industry point-of-view ........................................................................... 12

3.3 Aim and Objectives ....................................................................................................................... 13

4 Method .............................................................................................................................................. 14

4.1 Objectives .................................................................................................................................... 14

4.1.1 Identify evaluation methods...................................................................................................... 15

4.1.2 Conduct organizational study .................................................................................................. 15

4.1.3 Compare evaluation methods................................................................................................... 16

4.1.4 Selection of evaluation method .............................................................................................. 16

5 Identifying evaluation methods ........................................................................................................ 17

5.1 Analytic hierarchy process (AHP)................................................................................................. 18

5.1.1 Method description.................................................................................................................. 18

5.2 Multiple-Criteria Decision Aid (MCDA) ....................................................................................... 20

5.2.1 Method description.................................................................................................................. 20

5.3 Goal Question Metirc (GQM) ....................................................................................................... 23

5.3.1 Method description.................................................................................................................. 24
5.4 The PECA process for COTS evaluation ........................................... 26
  5.4.1 Method description ........................................................................ 26

6 Organizational study ........................................................................... 29
  6.1 Preparation ......................................................................................... 29
  6.2 Interviews ......................................................................................... 30

7 Evaluation of methods ......................................................................... 32
  7.1 Analysis of organizational study ......................................................... 32
  7.2 Method evaluation ............................................................................. 34
    7.2.1 AHP .............................................................................................. 35
    7.2.2 MCDA ......................................................................................... 36
    7.2.3 GQM .......................................................................................... 36
    7.2.4 PECA .......................................................................................... 37

8 Selection of evaluation method ............................................................... 39
  8.1 Software qualities compatibility ......................................................... 39
  8.2 Choosing a method ........................................................................... 39

9 Conclusion ............................................................................................ 44
  9.1 Summary .......................................................................................... 44
  9.2 General applicability of evaluation technique .................................... 44
  9.3 Deliverables ...................................................................................... 45
  9.4 Final analysis .................................................................................... 46
  9.5 Future work ...................................................................................... 46

References .............................................................................................. 47

Appendix ................................................................................................. 49

List of acronyms ....................................................................................... 49
1 Introduction

The purpose of this thesis is to examine software evaluation methods and their application in the domain of information systems (IS) development and business critical software in particular. The information system area of research is vast and encompasses many fields of research, among them the development and use of software. Today companies and organizations use software for many activities. The transition from “pen and paper” to computers has happened in a very short time and software has quickly become an integral part of daily operations, in many cases making them absolutely critical to the organization. Despite the importance of reliable and correctly functioning software, evaluation of IS and software products is often neglected (Bernroider and Stix, 2005). The area of software evaluation has been a rather popular topic of research and thus, quite a few methods, techniques and frameworks have been proposed over the years. However, choosing methods and applying them in the way it was intended is a difficult task where the outcome greatly depends on the experience and knowledge of those set to evaluate the software. Factors such as domain context, organizational structure and type of software further increase the level of difficulty when it comes to choosing the most appropriate method for the task at hand.

In this thesis, the aim is to investigate what software evaluation method is best suited for evaluating a specific kind of software in a real-world example. The software in question is a so called Intellectual Property Management System (IPMS) and the real-world example is in this case Volvo Technology Corporation (VTEC). This IPMS software falls under the category of business critical software, in that it is critical to VTECs daily operations. By first examining the organization to acquire knowledge of what the most important requirements are and which specific software characteristics are of greatest interest, we will be able to evaluate which of the described methods is most applicable. The goal is that the findings of this thesis and the thesis itself will be of some interest to evaluators of business critical software as well as to management of companies planning to invest in an information system.
2 Background

In this chapter, background knowledge and definitions of relevant concepts will be presented, giving the reader the necessary introduction into some of the topics that are pertinent to this thesis.

2.1 Information systems and organizational levels

In today’s information society, information systems (IS) of different kinds become a more and more integral part in companies and various business sectors. Almost all types of businesses, from the local gas station to the multinational car company, utilize IS’s on a daily basis. The term “information system” and what it envelops can be rather difficult to explain because of the sheer size of the subject matter. Stair and Reynolds, (2008) define it as a set of components in relation to each other that can gather, manipulate, store and spread data and information which provides a feedback mechanism to reach an objective. With this very basic definition it can consequently be argued that all software that has one or more of the above mentioned characteristics can to some extent be regarded as an IS. Furthermore, there are several types of IS (see Figure 1), each belonging to a specific organizational level of hierarchy. At the operational level we find transaction processing systems (TPS) whose databases often serve as foundations for high-level systems. At the mid-level of an organization so called “support of knowledge work”-systems are utilized. These typically include office information systems, such as groupware, professional support systems and knowledge management systems.

![Figure 1 – Levels of IS and systems inherent to each level of an organization](image-url)
At the upper-level of an organization, Management support systems are found, often related to decision support systems and executive information systems. IS at this level are of a more complex and advanced nature aimed at providing assistance to high-level executives to monitor the performance of the company, assess the business environment, and develop strategic directions for the future (Encyclopædia Britannica, 2009).

2.2 Document and information handling systems

Many office information systems today feature functions such as information retrieval and document management in order to further automate and optimize common office tasks. In essence, the objective of such features is to substitute documents that used to circulate in paper form with an electronic version and by doing so making the handling of documents more effective (Zantout and Marir, 1999).

The type of software that this thesis focuses on has modules to handle, what is usually referred to as reference documents, that represent a static source of information such as templates and pre-authored documents. The software also has the capability to retrieve and automatically edit these documents, for example filling out contact information, due dates, fees, etc. These are tasks that, if done manually, are very time and resource consuming, as well as tedious for the personal.

2.3 Application software & IPMS

One of the most common actual deliverables of an IS project is what is usually referred to as application software. Hoffer et al. (2005) define application software as software designed to support a specific organizational function or process. There are other definitions that are just as true but have a broader contextual view of the concept, i.e. Encyclopædia Britannica (2009) defines it as programs designed to handle specialized tasks, often sold as ready-to-use packages such as word processing programs. However, in the context of this thesis Hoffer’s definition is more relevant. The type of software that is in focus of this thesis is a so called intellectual property management system (IPMS). In essence, an IPMS could be described as a document and information handling system, designed to handle information of a intellectual property (IP) nature. The system is used for administration of intellectual property rights such as patents, brands and design patterns. An important part of this is the handling of the numerous documents and forms being sent back and forth between different instances and stakeholders. The IPMS can automatically generate such documents from templates stored in its database. Another important key feature of an IPMS is to keep track of terms and critical due dates for activities such as patent applications and renewals. It is difficult to place an IPMS in one of the levels of information systems discussed in chapter 2.1. One could argue that an IPMS in fact spans all three levels since it is utilized by many types of employees ranging from patent administrators working primarily at an operational level, to a strategic level where senior management can use aggregated data from the system for planning and decision making purposes. A multi-level system, such as an IPMS, ideally uses different levels of abstraction to achieve these level transgressions.
2.4 Business critical systems

According to Orci (1995) the definition of a critical system is one that has to function correctly in order to guarantee the organization's survival by avoiding loss of life or damage to health and environment. A system is also deemed critical if it, by functioning correctly and error-free, avoids great financial losses for the organization. Moreover, there are three categories of critical systems: business critical systems, security critical and safety oriented systems (Orci, 1995). Business critical systems are of great importance to the organization's daily operations and can be further categorized into, for example, administrative and operational systems. Examples of administrative business critical systems are workflow systems, billing and invoice systems and other such systems that the organization relies on in order for it to be able to function and take revenue.

By this definition of a business critical system, it can be argued that many systems and software products companies use on a daily basis are indeed of a business critical nature in that the company would not survive, in its current form, without them.

2.5 Failures of IS

For companies planning to invest in an IS there are many questions and problems that have to be properly dealt with in order to make a so informed decision as possible and to avoid the many risks and pitfalls that are inherent to IS investments. Ever since the birth of IS as we know it today, IS projects have been a highly risky enterprise to take on for an organization. In fact, there is a significant body of evidence to support that as many as 70% of all major IS implementation projects end in failure (Pan et al. 2009). A project is deemed as a failure when it has been terminated prematurely and without reaching the goals set out for it. However, a project is also deemed a failure should it not fulfill the original functional objectives once it has been developed and implemented. Such an IS project may be far more costly for an organization than a project failing completely but doing so at a relatively early stage. Some would argue that one of the major factors for failing IS projects are directly attributed to the growing complexity of today’s IS (Pan et al. 2009). Despite the apparent high risk associated with IS projects, the demand is growing as systems and software more and more become a natural part of organizations and individuals daily life.

As a natural consequence of this great demand for IS, the amount of software solutions on the market is vast and ever growing. This puts the buyer, i.e. the company that seeks a certain type of software for some specific task or area in their organization, in a situation where they may have to choose between a number of available alternatives, such as commercial-off-the-shelf (COTS) products, or consider an in-house solution.

2.6 IS investment evaluation

Firstly, it must be evaluated whether or not the proposed IS investment is relevant to the organization, i.e. does it satisfy some particular business need, increase the efficiency of certain aspects of the organization or simplify the execution of a task. This requires applying an appropriate method or mode of procedure which in itself presents new challenges in choosing the right method. The reality is, however, that proper evaluations are rarely done although there is a considerable amount of
literature that addresses common questions in relation to investments in new systems and software products (Bernroider and Stix, 2005). Many of the problems stem from difficulties in understanding the many complex factors that are involved in evaluating software systems, such as the span and limits of the system, its effects on the organization, the systems pros and cons, its inherent costs and risks and other strategic and potentially political consequences (Bernroider and Stix, 2005). Evaluation of IS investments is by no means an unexplored area in the realm of IS and much research on the topic has been conducted over the years. This has generated a large number of evaluation techniques and approaches as well as method surveys and comparisons. However, despite this extensive body of research studies have shown that in practice business management often fails to consider available IS investment evaluation techniques (Bernroider and Stix, 2005). Companies looking to invest in new IS commonly skip the crucial evaluation step altogether, especially medium to small sized companies. In cases where some sort of evaluation does take place, it is often limited to more widely understood accountancy approaches such as methods for discounted cash flow analysis (Bernroider and Koch, 2001). These sorts of evaluations are of a more generally applicable nature and can be applied to most corporate investment proposals. Because they are so commonly used and understood by senior managers they are often favored over more IS-specific evaluation methods. In cases where evaluation methods that are not explicitly financial in nature are applied, most likely they are of a simple scoring and ranking technique (Remenyi et al. 2000). Although such techniques may have certain advantages, such as being rather transparent and easy to follow, they are unable to capture and evaluate the full consequences of an IS investment. Furthermore, their success is almost entirely dependent on what criteria’s are chosen to be included (Remenyi et al. 2000). Again, in order to make the most informed decision possible, a proper evaluation of the IS investment must be conducted. In choosing methods for evaluating the investment it is important to have in mind that there are also a number of factors unique or closely linked to the organization in question that are just as important to consider. Therefore, applying methods that evaluate an IS investment based upon the organizations structure and particular needs, is of great importance.

### 2.7 Evaluating an IS

Once the decision has been made that a new system or a piece of software should be procured, the real question is which of the available solutions on the market fits the best? This is often no easy task and a whole new set of problems and challenges arise. Given the difficulty in choosing between software products, there is a substantial body of work related to defining frameworks and methods for product evaluation. The purpose of these methods is to give the “buyer” of the system, or rather the personnel involved in procuring a software product, the tools to evaluate available market alternatives and thereby hopefully choose the best solution for the task. These methods have different approaches as to what is to be evaluated in the product as well as how it should be evaluated. As with IS investment evaluation, one of the first major challenges is to choose the “right” method for the task.
2.8 Evaluating quality of an IS

As stated above, there are many methods for evaluation, each focusing on different sides of what appears to be the same coin. There are several wide-ranging frameworks and methods for evaluating the quality of an IS without them being limited to any particular category of software. These methods appear to be applicable to almost any form of IS related software, thus they have a broad scope and are generally less specialized than other techniques. Frameworks designed to evaluate the quality of an IS need to be based upon a holistic definition of IS quality, in other words one that encompasses all relevant factors including technological and organizational contexts so that specific needs of the company or organization are not excluded in the evaluation (Lamouchi et al. 2008). It also has to be taken into account that different types of organizations have different needs and characteristics which affect what types of evaluation criteria should be employed. Consequently, deciding what properties and criteria to include in the measurement must be considered as vital issues in evaluating IS quality. Examples of properties often included are usability, maintainability, reliability, portability and so on. When the environment, i.e. the company or organization in which the evaluation is to take place, has been analyzed a quality model links together and defines the various software metrics and measurement techniques that are to be used in the evaluation. Additionally, the approach taken must be sufficiently general for hybrid hardware and software systems (Lamouchi et al. 2008).

Although IS quality evaluation has been around for several decades, Wong and Jeffery (2002) claim that research in this area is still immature making it difficult for a user to evaluate software quality of a product. They state that there is still no clear definition of what software quality is, in part due to the wide range of interpretations of quality and the many aspects related to the concept, but also in the lack of consensus between what a user may regard as important quality characteristics and what a developer might read into the word.

2.9 The ISO Software Quality Model

Software quality itself is defined as a set of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (Losavio et al. 2003). The ISO 9126-1 standard specifies a hierarchical model of software quality at a relatively high level of abstraction. This ISO standard has become widely adopted for software architecture purposes and is now a software industry standard (Losavio et al. 2003). The quality model, as stated by the ISO standard, is at its highest level of abstraction constituted by six main software characteristics related to quality:

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability
By these characteristics or attributes, the overall quality of a software product can be both described and evaluated. A brief description of the characteristics follows:

*Functionality* bear on the software product's capability to provide functions that stand up to stated or implied needs, when used under pre-specified conditions. In other words, functionality refers to the ability of the software to provide appropriate functions to the users in order for them to successfully accomplish tasks.

*Reliability* is the software product's ability to uphold a certain level of performance under a stated period of time, under previously stated conditions. What this means in more practical terms is that the software must be able to continue to function correctly over a certain time span and without becoming i.e. unresponsive or unstable.

*Usability* refers to the software product's capability to be correctly used, learned and understood by its users under pre-set conditions. This characteristic encompasses a broad range of factors related to the relation between the software and the user, specifically how the user is able to understand and interact with the software in a desirable manner with a reasonable amount of effort.

*Efficiency* of a software product is directly related to its ability to deliver appropriate performance in relation to its resource needs. The software must be able to uphold a level of performance while not exceeding the resource needs as stated in previous conditions.

*Maintainability* refers to various factors related to the software product's ability to support modifications. This includes activities such as corrections and improvements as well as adaptations of the software that may be required in order for it to continue to be usable if organizational or contextual changes in the environment happen.

*Portability* evaluates the software product's capability to be transferred across environments, including hardware platforms, other software and also organizational transfers. There are several important factors to consider related to portability, data integrity being one of the most crucial.

The six characteristics are the foundation for the ISO 9126-1 model. However, to properly understand and be able to utilize this quality model, the main characteristics need to be further broken down and described. Losavio et. al. (2003) uses this model as the base for their quality requirements specifications and extends it with sub-characteristics that contribute to satisfying the main characteristic.

Functionality is extended with four sub-characteristics:

- *Suitability* refers to having the correct functions in the software so that the required tasks can be accomplished. For every appropriately specified task there must be a function that will satisfy that task.
- *Accuracy* means that the results or effects of the function must be in accordance with an agreed-upon degree of precision.
- *Interoperability* is the ability of the software product to interact and co-exist with other specified systems.
• Security refers to the software having some function, either hardware or software, that will prevent unauthorized access of data.

Reliability is extended with three sub-characteristics:
• Maturity is the software products ability to prevent or avoid failures caused by faults in the software itself.
• Fault tolerance refers to the ability of the software to maintain a certain level of performance if software faults occur. For example by having modules in the software dedicated to error and exception handling or by means of redundancy.
• Recoverability translates into three parts; the ability of the software product to re-establish a certain level of performance, the ability to recover potentially lost data and the time and effort needed to accomplish these tasks.

Usability is extended with three sub-characteristics:
• Understandability is the ability of the software product to be adopted and understood by the user. It also includes the products ability to enable the user to assess the overall suitability of the software and how to use specific functions.
• Learnability refers to how well the software product does in enabling the user to learn its application.
• Operability refers to the products ability to enable the user to control and use the software.

Efficiency is extended by two sub-characteristics:
• Time behaviour, in reference to system performance, is the ability of the software to function in a timely manner, specifically measured in response time, processing time and data throughput rates. These attributes apply to all functions of the software product and can be evaluated independently.
• Resource utilization refers to the software products resource needs for its different functions. Amount of resources needed and type of resource as well as duration of utilization are factors here.

Maintainability is extended by four sub-characteristics:
• Analyzability is the capability of the software to be analyzed for potential weaknesses or causes of failures. The ability to identify these parts of the software is the main focus when evaluating this sub-characteristic.
• Changeability is the products ability to support the implementation of pre-specified modifications.
• Stability refers to the software products capability to avoid malfunctions or unexpected errors as an effect of modification done to it.
• Testability is the ability of the software to be tested for faults, i.e. as a result of modifications.

Portability is extended by four sub-characteristics:
• Adaptability refers to the software products ability to be adapted to and utilized in other environments. For example, how well the product supports being used on another software- or hardware platform.
• **Installability** is the software products capability to be successfully installed in specified environments.
• **Co-existence** refers to the software products aptitude to co-exist alongside other software in the same domain, sharing common resources.
• **Replaceability** in concerned with the products capability to replace another similar system, i.e. taking its place, in the same environment. Both adaptability and installability are factors to be considered here.

These sub-characteristics are of varying importance for the quality of the software product, depending on the system in question (Losavio et al. 2003). The characteristics and their related sub-characteristics are illustrated in Figure 2.

The ISO 9126-1 standard provides little in terms of instructions or guidelines on how to customize the quality model. To successfully utilize the model one has to be aware of the properties that are expected from the architecture or generic framework on which the software system is to be implemented, as Losovi et al. (2003) points out in their article on specification of quality requirements for software architecture.

![Figure 2 – The ISO9126-1 Software quality model (Based on Losavio et al. 2003)](image)

### 2.10 Commercial-off-the-shelf (COTS) evaluation

When evaluating so called commercial-off-the-shelf (COTS) software products other factors come into play which can make evaluation even more difficult. The definition of COTS products and COTS-based systems seem to vary somewhat, making it difficult to give a clear and universally accepted definition. Comella-Dorda et al. (2004) defines a COTS product as one that has the following characteristics:

• sold, leased, or licensed to the general public
• offered by a vendor trying to profit from it
• supported and evolved by the vendor, who retains the intellectual property rights
• available in multiple, identical copies
• used without modification of the internals
However, Comella-Dorda et al. (2004) also recognize that many products are COTS-like, meaning they share some of the above mentioned characteristics but not all of them or share them to a certain degree. The authors also acknowledge that some of the characteristics themselves are difficult to define. For example, the statement that COTS products are used without modification to the internals brings up the question what the distinction is between tailoring and modifying a product.

Comella-Dorda et al. (2004) definition of a COTS-based system (CBS) is any system partially or completely constructed using COTS software products as integral components. They point out that the word “integral” as particularly important because it emphasizes that without the COTS-product in question, the system would not function.

Carney and Wallnau (1998) argue that COTS evaluation is notoriously difficult for a multitude of reasons. Firstly, the authors state that there is a lack of understanding of the process of software consumption. This is in part due to the wide variety of commercial software available and the very rapid pace in which new products become available. Moreover, the rate at which software products are further developed and refined makes classifying them even more difficult, adding to the problem of properly being able to understand them. Additionally, as a consequence of the diverse and quickly growing COTS market, it is unlikely that any single evaluation method employs an extensive enough criteria-basis to satisfy this variability. Carney and Wallnau (1998) explore this idea further by stating that because of the nature of COTS software, evaluation of these products may suffer from inherent incompleteness. An evaluator tasked with evaluating a certain product must first try to get an understanding of the many concepts and features the software claims to have. This in itself is no easy task since evaluation efforts are often hard pressed for both time and resources, making it hard for the evaluator to get a complete picture of the product. These efforts are further hampered by the fact that software is simply not predictable when it comes to for example interactions with other software. From a user perspective, software can be seen as a “black box”. Details concerning how the software works, the design and implementation, is often scarce for copyright reasons.

In conclusion Carney and Wallnau (1998) argue that given the difficulties stated above, trying to define and stick to a universal evaluation method for COTS software is an illusion and would result in over-generalized and imprecise evaluations that would be of limited use. Instead, the better approach would be to try to define a “philosophy” adaptable and comprehensive enough to be capable of dealing with the many variations of COTS software.
3 Problem statement

In this section, the problem that this thesis builds upon will be presented as well as the overall aim and objectives of the project.

3.1 Presenting the problem

The problem that is to be examined is which of the available methods and techniques for evaluation of software is most applicable on business critical systems based on organizational needs and requirements. The problem entails searching for and identifying relevant methods of software evaluation. At the first stage, picking out candidate methods based on literature and research in the information system domain. At a later stage, the organization (VTEC) will be studied in order to elicit requirements applicable to the specific type of system under review. The information obtained from this examination will then serve as a basis for evaluating which of the earlier identified methods is most inclined to provide the most relevant software evaluation results. A simple representation of the steps to be taken in this project is shown in Figure 3.

![Figure 3 – Model of the main objectives](image)

The research question derived from this problem statement is as follows:

*Based on an organization's specific requirements and dependency on particular software characteristics for a given situation, which software evaluation method is best suited for evaluating available software solutions?*

3.2 Why the problem should be studied

In this section the reasoning behind the problem and the justifications to why it deserves to be studied, are presented.

3.2.1 The need for evaluation

There has been an extensive amount of research done on information systems and software evaluation over the years. The need for evaluation of different kinds seems to be present at many stages of an information system project, from the initial investment proposal until the actual implementation and roll-out of the system. At all these stages, evaluation is key in order to maintain quality standards and prevent potential disasters that are inherent to IS projects. This thesis will focus on methods for evaluation of software at the selection stage, that is, when the organization in question have identified a need and decided to go about
choosing an appropriate system or piece of software that will fit their needs. Firstly, the need for qualitative and reliable software has been well established and the demand is constantly growing. Vlahavas, et al. (1999) argue that the requirements for software to comply with international standards and to be easily integratable with existing systems increase their complexity which also raises the bar for other parts of the software such as the user interface to maintain usability. Also, with increasing complexity come higher maintenance costs. Consequently, the need for proper evaluation is of great importance as software becomes a more and more integrated part of organizations and businesses today. For this reason, the effects of poorly evaluated software can be very severe, i.e. causing severe financial damage and loss of revenue.

3.2.2 Difficulty in selecting method

Since the area of IS evaluation has been a rather popular topic for several years there exists quite a few methods, techniques and frameworks concerned with comparing and evaluating different software solutions. In fact, because of the extensive assortment of methods, choosing the “right” method for the job is in itself a challenge. One could speculate that this, along with many other factors, might be a contributing reason to why so few organizations conduct a proper evaluation at the selection stage, just as many organizations skip evaluation at the investment proposal stage (Bernroider and Stix, 2005). The problem of choosing a method is in part due to that the applicability of different methods vary depending on the situation and the nature of the organization. A method that works well for one type of company may not work at all for another company because of organizational differences that the method in question may not be designed for or equipped to handle. Thus, studying the organizational context before choosing evaluation method is an important step.

3.2.3 The problem from an industry point-of-view

The aim of this thesis was developed and defined in cooperation with Volvo Technology Corporation (VTEC) in Göteborg. VTEC has a patent department in which they handle patents for many different parts of the Volvo group. This work used to be done manually but was computerized a couple of years back. The most recent system that was procured for this task is a so called Intellectual Property Management System (IPMS). This type of software is fairly extensive and it aims to automate the administration of many important and critical tasks related to intellectual property rights management. Among many other functions there is built-in document management, administration of important due dates and handling of contact information. The current IPMS has been in use for the last couple of years and at this point in time, VTEC has reached a crossroad where they are considering either buying a major upgrade of this system or procuring an entirely new IPMS. Should they decide to replace the current system, VTEC aims to scan the market for suitable software candidates in a proper manner. To accomplish this, some sort of evaluation method or technique that can assess important software characteristics has to be applied which of course entails first choosing the right method. From this situation, the problem statement of this thesis was formed.
3.3 Aim and Objectives

The overall aim of this thesis is to assess software evaluation methods and techniques that are applicable on business critical software. By doing this, the goal is to be able to identify a method of evaluation that will help the company, Volvo Technology Corporation (VTEC), in selecting the software solution best suited for their needs based on their organizational requirements. The actual deliverable this thesis aims to provide VTEC with is a suggestion of what evaluation method to use in order to compare market products and make a well founded decision as to which solution fits them the best.

The objectives are as follows:

- **Identify evaluation methods**
  - Identify a pertinent number of methods for software evaluation that are relevant to the study.

- **Conduct organizational study**
  - Examine the organizational needs and requirements in order to gain further knowledge about the current system and employees’ view of what characteristics are most important for a IPMS.

- **Compare evaluation methods**
  - Based on the organizational knowledge elicited from the previous objective, compare the selected methods.

- **Selection of evaluation method**
  - Based on the comparison of the methods, reach a conclusion stating which method is best suited for the task.
4 Method

The objectives and the possible methods identified in order to achieve them are as follows:

1. Identify evaluation methods
   a. Search for methods and frameworks for evaluation of IS or software products

2. Conduct organizational study
   a. Interviews
   b. Survey
   c. Observations

3. Compare evaluation methods
   a. Using a pre-defined method
   b. Define my own method for comparison

4. Selection of evaluation method
   a. Analyze results obtained

4.1 Objectives

For each of the objectives outlined in the previous chapter, the author will now argue what methods are to be applied in order to achieve each one of them in a satisfactory manner. Figure 4 illustrates the objectives and what methods will be applied to achieve them.

![Figure 4 – Model of objectives and methods.](image-url)
4.1.1 Identify evaluation methods

The first objective is to identify a number of methods for software evaluation that are of a relevant nature to the study. Having discussed the objective with the supervisor as well as with VTEC, it was decided that four evaluation methods would be identified and assessed. The motivation behind deciding on this particular number is that evaluating more than four methods may cause problems with the time-constraint of this project. The way this will be done is by searching for articles and research papers in bibliographic databases as well as journals and conference proceedings databases using combinations of keywords from the problem statement and aim, as described in Berndtsson et al. (2008). By doing so, the intention is to identify material relevant to the subject of software evaluation methods and techniques. By also looking at the reference lists of the articles found, additional material may be recognized that will further extend the findings.

4.1.2 Conduct organizational study

My second objective is to conduct a domain survey of the organization, which is Volvo Technology Corporation in this case. In order to gain information on the current IPMS and to get an understanding of the environment in which the software in question is to be used, information has to be gathered so that the methods identified in the previous objective can be properly evaluated. There are several ways of gathering information of this kind, such as interviews, surveys, observations and even case studies. In this case, we are primarily considering using interviews as it is believed this method is most applicable and will yield the best results, which will be argued for in the section below.

By interviewing people in charge of the patent department as well as the personal working with the system on a daily basis, the ambition is to gain knowledge of what software characteristics are the most important ones. The reason interviews are favored over, for example, surveys, is that surveys are more appropriate when there is a large amount of respondents. In this case there are quite few persons with relevant knowledge of the issue, making interviews a more suitable method. Additionally, investigating more complicated issues is very difficult using surveys as there is no two-way communication between the interviewer and the interviewee (Berndtsson et al. 2008).

Conducting observations is also a technique that could have been used, however, in this case the author believes it would be a poor substitute for interviews. Although observing users interacting with the system will provide some insight into how the system is perceived and used, it would not provide the same level of detail and understanding that interviews hopefully will. That said, observations would serve as a good compliment but considering the narrow timeframe of this project, it is unlikely there will be time for proper observations.

Another contributing factor to why the author believes interviews is a good option is that a rudimentary level of trust has been established with some of the people involved in that the author have met, and also to some extent worked, with them on a previous occasion. Since the quality of the results from an interview is so heavily dependent on trust (Berndtsson et al. 2008), this should help in sustaining quality of the information collected. There is, however, the question of bias in cases such as this.
The fact that the interviewer have pre-established relationships with some of the personal is a potential disadvantage and risk since this could “colour” the results and the interpretation of these. Eliminating bias is very difficult since it is a psychological matter and thus cannot be completely controlled. However, by thoroughly describing the background and basis in regard to this, the question of bias should be sufficiently accounted for (Berndtsson et al. 2008).

4.1.3 Compare evaluation methods

The third objective is to compare and evaluate the methods identified earlier. There are basically two ways of doing this; either by using a pre-defined and established method for evaluation, or by defining my own evaluation procedure. Having considered both methods and consulted knowledgeable people in the area, it seems that the second method is the better choice. Since the end-goal is to select a software evaluation method that will best suit the needs of the company, the methods have to be compared and evaluated on the basis of the organizations requirements which were elicited in objective two. The search for a method for evaluation of other methods that incorporate the requirements stated above have not been successful, and therefore a mode of procedure is favored where the author carefully describes how the evaluation of methods will be conducted and letting that serve as a method in of itself. What is aimed to be accomplished in this step is to extrapolate information from interviews and documentation that allows me to pinpoint what software characteristics and requirements are most important. We will then proceed to evaluate the methods suitability with the identified requirements and characteristics as basis for the evaluation.

4.1.4 Selection of evaluation method

The fourth and last objective is to reach a conclusion regarding which of the compared methods will best suit the organization and the problem described. This requires analyzing the results obtained from the method evaluation in a proper manner so that the conclusions reached can be considered sound and valid. The author plans to do this by carefully documenting the progress in analyzing the data, so that the reasoning can be followed and evaluated.
5 Identifying evaluation methods

One finds, when searching for articles and papers about software evaluation, that this is no new area of research in the field of IS. On the contrary, there is a substantial amount of scientific articles, conference proceedings and research papers dealing with various aspects of information systems and software evaluation. Already in the 1960s, researchers began working on IS related evaluation issues. Since then, IS evaluation has become one of the most researched and written about topics in IS research, resulting in a large number of evaluation techniques available today (Bernroider and Stix, 2005). Considering this vast array of information on the subject, in order to identify relevant sources of information for this thesis, the searches had to be properly set up.

By using combination of key words from the problem statement and aim, bibliographic databases were systematically searched through. Primarily, combinations of the words; “software”, “evaluation”, “methods”, “information systems” and “business critical” were used.

At this early stage of searching for methods, “broad” search words were deliberately used as this was expected to return many results so as not to exclude potentially pertinent articles. The main objective at this stage was to identify the main disciplines in software evaluation from which more refined and detailed techniques and frameworks may have spawned.

As expected, this yielded many results in all databases that were consulted. By sorting the results after relevance, each result was examined, attempting to determine if it was relevant to the study based on the title and abstract of the article. In many of the search results it was obvious by only reading the title that they were not even remotely relevant. Some results might have been of some relevance but were instead aimed at different fields of research such as biomedicine or healthcare, rendering them less useful for the purpose of this thesis.

Articles deemed to have a good chance of being of interest were then retrieved and sorted as “A”-results, while articles that might be of interest were sorted as “B”-results. This procedure was repeated for several bibliographic databases including “ScienceDirect”, “ACM Portal”, “SpringerLink” and a few others.

Having done this initial selection, a group of relevant articles and papers dealing with software evaluation had been compiled. By going through the material in more detail, it soon became increasingly clear which of the articles could indeed be deemed relevant and which were non-applicable for this thesis. Since the objective for this stage in the thesis was to identify four methods for software evaluation, a secondary selection had to be conducted among the batch of candidate methods in order to reach that number. The main factors that were considered in this last phase of selection were primarily to which extent the methods seemed well established in the research community and if they were generally utilized and accepted for evaluation purposes.

While conducting this search for methods it became increasingly clear that because of the huge amount of research material, it is quite possible that potentially relevant methods were not identified in the investigation. Given the time constraints on the thesis project it is unrealistic to expect that all areas can be covered and thus, no
assurances can be made that all available methods that might have been of interest to the study were considered.

The methods finally chosen for evaluation in this thesis will now be presented and motivated along with an introduction and description of their application.

5.1 Analytic hierarchy process (AHP)

The analytic hierarchy process is one of the most researched and written about methods for multiple criteria decision making (MCDM). It was first developed by Saaty (1980), and has since then been the focus of many scientific articles. Since its inception, the AHP method has proven to be applicable to a very wide range of decision-making problems, as stated by Vaidya and Kumar (2006) in their extensive literature review of 150 application papers of AHP. Examples of classic applications of AHP tools within the decision-making realm are planning, selecting a best alternative, resource allocations, resolving conflict, optimization, and numerical extensions of AHP, just to name a few. Furthermore, AHP has been adopted and utilized in many different fields such as education, engineering, industry, government, manufacturing, management, political, social and even sports (Ho, 2008). The remarkable success of AHP is partly due to the simplicity of the method itself along with its ease of use and relatively flexible nature. Considering the established position that AHP has in being an often utilized tool for evaluation and decision making purposes, it has been chosen as one of the methods to be evaluated in this thesis.

5.1.1 Method description

Simply put, AHP is a tool that uses well defined mathematical structures to determine which alternative from a set of candidates is the best one. Thus, it is a mathematical approach to a given decision making problem but it also incorporates and builds upon subjective input from the personal involved in the decision making process.

Ho (2008) gives a more technical description of the method where he states that the AHP consists of three main operations, those being:

- Hierarchy construction
- Priority analysis
- Consistency verification.

In the first operation the decision makers are required to break down complex multiple criteria decision problems into, what Ho (2008) refers to as, its component parts. Every possible attribute of the parts are then arranged into multiple hierarchical levels.

This is then followed by the priority analysis at which time the decision makers compare each cluster in the same level in a so called pairwise fashion. They do this based on their own experience and knowledge. For example, every two criteria in the second hierarchy level are compared concurrently with respect to the goal, whereas every two attributes of the same criteria in the third level are compared concurrently with respect to the corresponding criterion. Since these comparisons are conducted on the basis of the decision makers’ personal or subjective judgments, a certain degree of inconsistency may occur. This is natural and part of the process but to guarantee that
the judgments are consistent the third and last operation, consistency verification, is performed. This operation is regarded as one of the most important and advantages of the AHP. Its purpose is to incorporate a certain measure of consistency among the previously pairwise compared sets by calculating the consistency ratio. Should the operation reveal that the consistency ratio exceeds that of a pre-defined limit, the decision makers should investigate and possibly revise the pairwise comparisons. Once all pairwise comparisons have been conducted and have proved to be consistent, the judgments can be synthesized in order to elicit the priority ranking of each criterion and its subsequent attributes. The overall procedure of the AHP is shown in Figure 5.

To explain the AHP in a more clear and understandable way one could attempt to break down the process in stages.

Stage 1: There are three main parts that need to be accomplished. The overall problem or objective needs to be formulated, i.e. “Choose a new system”. The criteria for this objective needs to be defined, i.e. functionality, usability, maintainability etc. Thirdly, alternatives need to be stated, i.e. relevant systems that are available on the market. These parts are then arranged into a hierarchy, much like a hierarchy tree-chart with the main problem at the top, followed by its derived criteria and the alternatives identified.

Stage 2: The criteria now need to be synthesized in order to determine the relative ranking of the alternatives. At this stage the decision makers use their judgment to determine the rankings. For example, making the judgment that “usability is 3 times as important as portability” and “portability is twice as important as maintainability”.

![Figure 5 – The AHP model (Based on Ho, 2008)](image-url)
Stage 3: At this stage the pairwise comparison of criteria is executed in order to elicit the relative importance of one criterion over another. The criteria are arranged into matrixes and assigned values that indicate for every one criterion, its importance against the other criterions. The scale ranges from 1/9 for “least valued than”, to 1 for “equal” and to 9 for “absolutely more important than” (Vaidya and Kumar, 2006).

Stage 4: When the pairwise matrix is completed and all criteria have been assigned values against each other, the matrix needs to be transformed into a ranking of criteria. This is done with the utilization of the “eigenvector”. In brief this entails converting the fractional values of the criteria to decimal form and raising the values to powers that are successfully squared each time. The sum of each row in the matrix is then calculated and normalized until the difference between the sums in two successive calculations is less than that of a pre-defined value.

Stage 5: The computed eigenvector values produce a relative ranking of the criteria which can now be inserted into the hierarchy tree.

Stage 6: Stage 2-4 is now repeated for the alternatives defined in Stage 1. The decision makers judge each alternative until a relative ranking is determined. They then proceed to perform a pairwise comparison, weighting each alternative against the other. Finally the eigenvector is used to transform the matrix into a ranking of alternatives.

Stage 7: The calculated value for each alternative is inserted into the hierarchy tree. To reach a conclusion as to which alternative is the best, some final calculations need to be performed. Each alternatives criterion value is multiplied with the overall criteria value. These numbers are then added resulting in a final value. The alternative with the highest value is the highest ranked alternative.

5.2 Multiple-Criteria Decision Aid (MCDA)

The MCDA methodology is a very useful technique for performing evaluations of any kind. Vlahavas et al. (1999) describe MCDA as a methodology aimed at evaluation problems where the final result depends upon on many criteria. With such a broad definition one could reason that MCDA is potentially applicable to almost any situation where a decision has to be made when choosing between \( N \) alternatives. However, to successfully perform an evaluation, one must first select a number of attributes which in turn can be associated with some form of measure or metric, either directly or indirectly. This step is one of the most crucial in an evaluation process, as the chosen attributes and the means by which they are to be assessed, reveal the focus or standpoint of the entire evaluation (Vlahavas et al. 1999). MCDA itself has spawned several other evaluation methods that use and build upon fundamental MCDA principals.

5.2.1 Method description

A software evaluation problem that uses Multiple-Criteria Decision Aid can be modeled as a seven-step procedure, as outlined by Vlahavas et al. (1999), see Figure 6:

- \( A \) is the set of alternatives under evaluation in the model
Step 1: First, a number of alternatives $A$ must be defined, i.e. if the problem is which software system to choose, the candidates have to be identified. In the unlikely event that only one alternative is to be evaluated, a prototype has to be defined that fulfills certain requirements. The alternative and the prototype are then compared. This first step can be considered a first level evaluation in which certain alternatives may be disregarded depending on if they fulfill specific requirements or not.

Step 2: Before the actual evaluation takes place the type $T$ of desired outcome must be determined. There are several possible outcome types, each serving different purposes. Depending on the evaluation problem the results of the evaluation can take different forms. One possible result is a set of choices, where the alternatives are sorted into subsets of “best choices” and “less desirable choices”. Classification is another possible result where the alternatives are again sorted into subsets that have a classification such as “best”, “good”, “worst” etc. Sorting is perhaps the most commonly opted-for evaluation result. This means a ranking of the alternatives from best to worst choice. Another result is a description of the alternatives without any ranking.

Step 3: In this step, considered to be the most important one in MCDA evaluation, the decision makers must define a set of attributes $D$ to be evaluated and their hierarchy. A visual representation of this is a hierarchy-tree where the entity to be evaluated is at the top level. Below, attributes are inserted and broken down into sub-attributes until they reach a basic form and cannot be divided further. Attributes that can be broken down are called compound attributes. The attribute tree defined in this step indicates what direction the evaluation will take and it is therefore important that this process is carefully thought through. Additionally, there are two different approaches that the evaluation can take from here (Vlahavas et al. 1999). The fixed-model approach is a more easily managed and simplistic model where the set of attributes is definitely identified and tailored to a specific evaluation problem and domain. The drawback of the fixed-model is its apparent lack of flexibility. The second model approach is called constructive model. Unlike the fixed-model, the set of attributes in the hierarchy tree is pre-defined for the type of evaluation problem. Attributes may be added or removed, allowing the tree to expand or reduce giving it high degree of flexibility. However, this approach requires skill and experience from the user. Another risk with the constructive model is that redundancy may be introduced among the attributes, potentially resulting in erroneous outcomes.

Step 4: For every attribute a certain measurement method $M$ must be defined. There are two different types of values, namely arithmetic values where the values are represented by numbers, and nominal values where descriptive values such as “best”, “good” and “least good” are used. Should the measurement prove to be impractical for an attribute, this attribute may need to be assigned an arbitrary value. Alternatively the attribute could, if possible, be broken down further into components that are measurable. However, both methods introduce a degree of subjectivity.
**Step 5**: For every basic attribute a measurement scale $E$ must be chosen and defined. If the measurement method chosen in step 4 was an arithmetic value, the scale usually follows the type of metric used. For nominal values the evaluator declares the scale to be used. Also, the scales must be of so-called ordinal nature, that is it has to be clear which of any two values implies the highest rating.

**Step 6**: This step involves defining a set of preference structure rules $G$. What this means is that for every attribute and its associated measure, a rule must be defined that can transfer measures to preference structures. These structures, called basic preference structures, compare a set of two alternatives based on some specific attribute. Basic preferences can be combined creating global preference structures.

**Step 7**: The final step in the basic MCDA methodology is the selection of a suitable aggregation method $R$, which is an algorithm that transfers the set of preference relations into a so-called prescription on $A$. There are different types of aggregation methods in the MCDA methodology that each belong to one out of three classes:

- The multiple attribute utility methods
- The outranking methods
- The interactive methods

The selection of an aggregation method depends on various factors such as:

- Type of problem
- Nature of the set of possible choices
- Measurement scales chosen
- Type of parameters or “weights” connected to the attributes
- Dependencies among the attributes
- Type of uncertainty present

![Figure 6 – MCDA basic evaluation procedure.](image)

Finally, the implementation of these seven steps does not have to follow this specific structure. For example, $D$ can be defined before or in parallel with $A$. 

22
5.3 Goal Question Metric (GQM)

A commonly used method for IS quality evaluation is the GQM method (Goal-Question-Metric) which has been under development by NASA since it was first described by Basili (1984). According to Wong and Jeffery (2002) the most notable feature of the GQM, is the central role of a goal. If the selection of characteristics and properties being measured are not clearly motivated it is likely that the data collected from the evaluation will be unfocused and of inadequate relevance. Moreover, Wong and Jeffery (2002) argue that the GQM-method provides little in terms of help or guidelines on how it is to be used, instead relying on the experience and competence of the personal involved in the evaluation.

This focus on goals and organizational key factors differentiates the GQM approach from other decision making methods. Specifically, before the method can be applied, the organization must first specify its goals where after it must specify the data that is intended to define those goals (Basili, 1992). This methodology is therefore goal-driven and the outcome of applying GQM to a certain problem is a specification on a measurement system that targets a particular set of issues and rules. This measurement model has three levels, as defined by Basili (1992):

**Conceptual level (Goal)**

Some particular goal is defined that identify what is to be accomplished relative to three basic measurement objects, namely:

- **Products** – Related to objects such as documents and deliverables that are developed during the systems life cycle.
- **Processes** – Activities related to system development such as designing, testing, implementing etc.
- **Resources** – Basically objects utilized in the development process of a system, for example hardware, software, personnel etc.

**Operational level (Question)**

For every goal a range of specific questions are formulated regarding how the goal is to be accomplished. The aim of the questions is to characterize the object/entity based on some specific quality issue and also to establish the objects quality from a certain viewpoint.

**Quantitative level (Metric)**

To be able to answer a question in a quantitative manner, a set of metrics are associated with every question. The metrics/data are of either objective or subjective nature.

- **Objective** – Data that depends just on the object being measured
- **Subjective** – Data that depends on the object being measured and the particular viewpoint from which they are taken.

The GQM methodology, like many other evaluation methods, is of an hierarchical nature. A goal is first defined and then described as a set of question. These questions
are then associated with metrics that can measure the object under evaluation. The metrics can be either objective or subjective (Basili, 1992).

5.3.1 Method description

The application of GQM can be described as a six step process:

**Step 1 – Establishing goals**

Firstly, various goals must be set for the organization. These are usually defined as either business goals or measurement goals. To differentiate these two types can be tricky as they are not always mutually exclusive. The main concern here, however, is to be careful where the goals are originating from and what focus they have. According to Basili (1992), the business goals should be defined based on the organization’s long-term plans for the project and overall future direction of the organization itself, rather than being limited to the particular scope of the current project. Well defined business goals are an absolute requirement for GQM to be of any practical use. Whether they are developed as part of the GQM process or strategic planning viewpoint is irrelevant. What is important to be aware of is that the measurement goals build upon the business goals and therefore the measurement program will lack focus and validity if business goals are not set.

In a GQM model the goals at the top are the result of Step 1. To define these conceptual measurement goals, the goal statements possess five types of information (Basili, 1992):

- **Object:** The product or process being evaluated.
- **Purpose:** Motivation behind the goal, why it is being examined.
- **Focus:** The quality attribute of the object under study.
- **Viewpoint:** From whose perspective is the goal developed.
- **Environment:** In what context and to what extent in the organization is the goal valid.

**Step 2 – Generating questions**

The purpose of defining question to the measurement goals is to clarify and further refine the goals to rid them of potential disambiguations. This advances the program by moving from the conceptual level of the goals in step 1, to an operational level. Additionally this process is an effective way of identifying different viewpoints on the goals within the project team.

The questions should ideally capture all the different sides of the goal so that it can be properly evaluated whether the goal is being met or not. In defining the questions, one of the most difficult aspects is finding the right level of abstraction contra detail. If a question is too abstract it may be difficult to see the connection to the metric used to measure it. On the other hand, a too detailed question may make it difficult to correctly interpret the goal on which it is based. Often, it is necessary to break down questions into sub-questions in order to reach a satisfactory level of detail. This also helps in identifying metrics for each question later on.
Step 3 – Specifying the measures

In this step the process is advanced from an operational level to a quantitative level. For every defined question, a metric need to be assigned that can properly answer the question. A metric is basically a means of measuring the level to which the question has been answered.

The stakeholders that are concerned with the object of the goal need to be directly involved in the identification of the metric to avoid misunderstandings and ambiguities, furthering the overall validity and consistency of the process.

Step 4 – Preparing for data collection

Moving further down the model, once metrics have been defined data items need to be identified that can support these metrics. The metrics themselves specify how the data needs to be organized in order to make sense to the recipient of the information. Constructing a schematic for how the data collection is to be performed is a rather immense and time consuming task that often benefits from a so called “Measurement plan”. Such a plan usually contains information such as definitions and descriptions on measurements, possible outcomes of the measurements, who is responsible for collecting each measurement, and at what time/frequency the measurement should be collected.

The main purpose of the measurement plan is to define and describe all different forms and types of data collection, as well as what automated data collection tools that should be used, if any. Additionally, it addresses the question of how the data can be collected most efficiently and effectively and to whom it should be delivered (Basili, 1992).

Before implementing the program a trial period is usually desirable so that potential weaknesses and oversights can be corrected.

Step 5 – Collecting, validating and analyzing data

Assuming the measurements plan specifies the data collection process, this step will have been pre-defined. Data collection is usually a reoccurring process done throughout the program and an integral part of GQM. However, simply collecting data is useless unless the data is actually used. Validation of the data must be done in order to confirm that the “correct” type of information has been gathered for each metric. Thirdly, the collected and now validated data must be analyzed. Analysis of data typically entails organizing the data and preparing the metrics for presentation to the stakeholders to address the questions pertaining to the measurement goal.

Step 6 – Analyzing the data for goal attainment

With the measurement goal fulfilled, this last step is analyzing the outcome of the process in order to learn from experience and mistakes.
5.4 The PECA process for COTS evaluation

The PECA (Plan, Establish, Collect, Analyze) methodology was developed by the Software Engineering Institute (SEI), a federally funded research and development center sponsored by the U.S. Department of Defense, at Carnegie Mellon University in collaboration with National Research Council Canada (NRC). The method, which is designed for evaluation of commercial-over-the-shelf (COTS) software, aims at helping organizations make informed and well reasoned decisions when procuring software products (Comella-Dorda et al. 2004). It is designed to be flexible so that a wide variety of organizations can use it and to allow it to be applied to many different kinds of COTS evaluation problems.

5.4.1 Method description

The PECA method is constituted by four main activities:

- Planning the evaluation
- Establishing the criteria
- Collecting the data
- Analyzing the data

As Figure 7 illustrates, the four activities do not necessarily need to be executed in a sequential manner. On the contrary, as with any real-world information system project, unforeseen and unplanned events inevitably occur in an evaluation process. The ability of PECA to support, for example the introduction of new criteria or unexpected events while collecting data, is one of its major strengths.

Figure 7 – The PECA evaluation process (Comella-Dorda et al. 2004)
**Step 1 - Planning the evaluation**

Naturally, the planning of an evaluation differs depending on the situation and type of problem at hand. Usually though, planning includes forming the evaluation team, creating a charter, identifying stakeholders, and picking the approach (Comella-Dorda et al. 2004).

Assembling a good evaluation team is a very important activity and its impact on the entire evaluation should not be overlooked. According to Comella-Dorda et al. (2004) an evaluation team could include:

- Technical experts
- Domain experts
- Contracts personnel
- Business analysts
- Security professionals
- Maintenance staff
- End users

The evaluation team, once formed, should then construct a “charter”. A charter is basically a range of project specific questions such as: “What is the evaluation expected to achieve” and “When should the evaluation finish”. Basic question like these are often overlooked and not documented which can potentially have disastrous results later on in the process if difficult situations arise. The charter should also include the goals and scope of the evaluation as well as the names and roles of the evaluation team, along with other fundamental factors.

Another crucial stage in planning the evaluation is picking an approach, meaning determining the primary characteristics of the evaluation activity. Picking an approach usually includes:

- Deciding the depth of the evaluation
- Determining the basic strategy for selection of an alternative
- Utilizing filters to reduce the number of candidate products

**Step 2 – Establishing criteria**

As with most evaluation methods, deciding on what criteria to use is considered the most crucial step in PECA (Comella-Dorda et al. 2004). Essentially, this step includes three activities:

1. Define evaluation requirements
2. Define evaluation criteria
3. Prioritize criteria

Before being able to define criteria, evaluation requirements must be defined. System requirements are written at an abstract level so that several technical solutions will be possible. Requirements should be stated as needs, for example “The system needs to be platform-independent”.
Requirements then need to be translated to actual measurable criteria. As Comella-Dorda et al. (2004) states, before commencing the evaluation of products, evaluation criteria must be generated from the evaluation requirements. Criteria are the specification on the practical approach chosen for determining whether a certain COTS product has the characteristics that are required for the system. While requirements are formulated as needs of the system, criteria should be formulated as abilities to satisfy the needs, for example the requirement of a platform-independent system could be transferred to a criteria stating: “The product should be programmed in Java or other platform independent language”.

**Step 3 – Collecting data**

Provided that proper requirements and criteria have been defined, the data collection process can begin. Data collection provides a basis for analysis. Collection should be simple, repeatable, and measure what it was intended to measure and capture information in a form suitable for analysis (Comella-Dorda et al. 2004). Accurate data collection is critical to a successful evaluation. Techniques employed for collecting data will vary depending on the nature of the evaluation. There are however a number of often used collection techniques such as literature reviews, vendor appraisals and various hands-on techniques. These will not be explored in detail here, but the expected results of data collection are important to mention. The overall goal is of course to learn how well the different candidate products stand up to the defined criteria. However, this is not necessarily the most important information. Data collection will also provide valuable information on the marketplace.

**Step 4 – Analyzing results**

This final step of PECA entails analyzing the results of the data collection phase. According to Comella-Dorda et al. (2004), data collection typically produces a large number of facts, checklists, and other types of data. For this information to be of any use, it must be consolidated into information that can then be analyzed. Comella-Dorda et al. (2004) describe several techniques for consolidating data including:

- Weighted aggregation
- Sensitivity analysis
- Gap analysis
- Cost of fulfillment

The last step of PECA is providing a recommendation based on the evaluation. This step is not as simple as just stating “Alternative A is best based on our results”. The information and results of the evaluation must be compiled and presented in a comprehensive manner since different stakeholder will be interested in different aspects of the evaluation results (Comella-Dorda et al. 2004).

In conclusion, PECA is perhaps more a guide for evaluating COTS-products than it is a clear-cut methodology. The four steps of PECA all offer different ways and methods for completing them, far more than can be written about in this thesis, thus providing great flexibility but also being less easy to follow than other more closely defined and structured evaluation methods.
6 Organizational study

This chapter describes the preparations and execution of the organizational study that was conducted.

6.1 Preparation

The objectives of conducting an organizational study is to, as stated in chapter 4.1.2, gain knowledge of what specific software characteristics are of greatest interest as well as eliciting the most important requirements of the system. The aim is not to compile a complete requirements specification, but to gather information of key aspects of the intended system so that this knowledge can then be used as a basis when evaluating the methods identified earlier. As also described in the “Method” chapter, the way this will be done is primarily by means on interviews. Having discussed the interviews with VTEC, a number of key employees with different work roles have been identified as the most appropriate persons within the organization to interview. They all use or come into contact with the system in question on a daily basis. Because of their different roles in the organization, they all have their own relationship with the system and thus have varying experiences.

The respondents include:

- Four administrators or “patent registrators” who work with the system on a daily basis
- One economist working with financial questions in the system
- One patent attorney who also has the overall responsibility for the system
- One employee from the IT-department, in charge of local technical support for the system

Regarding the style of interview to be used there are basically two different kinds to choose from; the open interview and the closed interview (Berndtsson et al. 2008). In an open interview, often used in qualitative research, the agenda is not set in advance and the interviewer has limited control over the session. Although the purpose of the interview is clear to the researcher, the topics brought up in the session is very much up to the interviewee while the researcher instead tries to guide or “steer” the focus of the interview into areas he/she thinks is especially important to the interviewee (Berndtsson et al. 2008). The main advantage of the open-style interview is that the issues of real importance may be identified since it’s the interviewee that is in control. The success of this type of interview is however largely dependent on the researchers proficiency and skill, introducing a rather large risk factor, should the researcher not be able to correctly steer the interview.

The second interview style is the closed interview, also called pre-structured interview (Berndtsson et al. 2008). As the name suggests, in such an interview the questions are formulated and set beforehand. It has the advantage of repeatability over the open style interview, making it more suitable in for example survey research.

In deciding which type of interview to use, one has to consider the appropriateness of the different techniques in relation to what kind of information the interview is
suppose to yield as well as the role of the interviewee. The respondents chosen for interview sessions in this organizational study have different roles in the company and thus the interviews have to be somewhat modified for each of the different roles. The type of interview chosen in this study is a hybrid of the open interview with elements from the structured interview approach. The questions were formulated in such a way that simple yes or no answers would be avoided. Instead, the intention was to open up a certain topic area with a question that would get the interviewee to elaborate on the answer and not feel constrained to only give an answer to the specific question.

6.2 Interviews

Although the objective was to gather information on requirements and characteristics of a possible future replacements of the current system, all the questions asked in the interviews, with the exception of a few, were related to the users experience and knowledge of the IPMS currently in use. By asking the respondents questions about their experience and perception of the system as well as asking them what features and functions they are happy and unhappy with, one can derive from the answers what characteristics are of highest importance for the IPMS.

The interviews were conducted onsite at VTEC. For practical reasons this was the only reasonable course of action as it would require the least amount of time and effort from the respondents. Additionally, as Berndtsson et al. (2008) points out, interviewing the respondents at their workplace will hopefully make them feel more comfortable with the situation and enable them to be more open and forthcoming in their answers.

The sessions were carried out in private, one interviewee at a time, in a group room. In order to help the flow of the interview and to make sure no potentially important information was missed, recording devices in the form of two voice recorders, were utilized. As the interviews were designed as a mix between the open- and closed-style techniques, using recorders was necessary since taking notes during the course of the interview would have been difficult. The downside with using voice recorders is that it may have a negative impact on the respondent. They may feel uncomfortable and regard the recorder as a distraction (Berndtsson et al. 2008) which could translate to less qualitative and exhaustive answers; however this was a necessary trade-off.

Before each session the interviewee was asked whether they approved the use of voice recorders. They were also informed of their rights to at any time discontinue the interview, retract answers either directly or after the session and of the fact that the interview was anonymous. No interviewee had any objection to the session being recorded or any other remarks about the interview itself. The interviewee would start by briefly explaining their role in the company and their relation to the system where after the actual questioning began. The interview questions themselves were designed with the specifications of quality requirements as stated in the ISO 9126-1 Software Quality model. For each characteristic including its inherent sub-characteristics, questions had been pre-defined that were of relevance to that specific software quality with the intention that the answers supplied by the respondents to these questions could be classified as belonging to that specific characteristic.

For instance, a “functionality question” was “Do you feel the current system is functional enough for you to be able to perform your work tasks properly?”
The answer to this question and the follow-up questions where the respondents were asked to elaborate on their answer could then be classified as relating to the functionality characteristic, provided that the answers stayed on-topic.

The interviews varied somewhat in length depending on who the interviewee was and his/her role in the company. The interviews with the patent administrators took approximately twenty minutes each, while the sessions with the head of the patent office and the employee from the IT-department took around forty minutes.

The issue of confidentiality is an important one to consider when conducting interviews. Preservation of confidentiality extends to all notes and recordings from the interview as well as transcripts of these. Also the results and information derived from the interviews presented in a final report must sustain the same level of confidentiality (Berndtsson et al. 2008).

In this case, the question of confidentiality was not of particular importance to the company or the interviewees. However, as already mentioned, the interviews were kept confidential and the transcripts from the voice recordings contain no names of the interviewees.
The process of evaluation of the chosen methods is based on a number of factors. As already mentioned in the Method chapter of this thesis, no pre-defined “method for evaluating methods” will be utilized. This is partly due to the fact that no such method has been identified, but also because of the desired outcome of this objective. The goal is to opt for one of the previously chosen and described methods that will serve the company best in helping them to select an IPMS solution. To elicit information on the organizations requirements and wishes on such a system, an organizational study in the form of interviews were conducted as described in the “Organizational study”-chapter. The analysis process and the results of these interviews are presented in the section below.

### 7.1 Analysis of organizational study

Once all the interviews had been conducted, the first step was to transcribe each recorded session. The transcription process was a fairly time consuming procedure. However, all audio recordings were of good quality and no problems were encountered during the process. To preserve anonymity no names are mentioned in the documents.

After having transcribed all interviews, the analysis of them could begin. As described earlier, the questions were designed with the ISO Software quality model in mind and thus, the answers would be evaluated and classified accordingly. Each of the six main software qualities, as described by Losavio et al. (2003), were associated with a color i.e. functionality was yellow, portability was red and so on. Every transcript was then analyzed by color-coding the answers and statements with the corresponding software quality color. For example, while talking about functionality and what functions the current system lacked, one interviewee remarked that:

“I am used to being able to print out lists of different kinds from the system. This is an important function especially in these times. Clients want to know how they’re doing.”

This statement was classified as being related to the functionality characteristic in the ISO model. Of course, far from all statements and answers were this simple to classify. Indeed, “decoding” and branding interviewees answers was one of the more difficult parts of the organizational survey and required a great deal of time and precision. Primarily, the definitions of the different software characteristics that Losavio et al. (2003) declare in their article on quality characteristics for software architecture were used. As the characteristics are quite extensively explored and described in this article, the author believes that the accuracy of the final results of the analysis is of satisfactory quality to the survey and this thesis as a whole. However, it stands to reason that factors such as experience bear on the outcome of an analysis of this kind and therefore a more experienced and knowledgeable person may well reach different results. Although, provided that the same analysis is performed under similar circumstances, the author believes that the results would not differ to any great extent.

Another factor that should be taken into account is that the author have had previous experience with the system in question and therefore is able to relate and better
understand many of the respondents’ remarks and comments than someone who is completely unfamiliar with the system. Although the author believes this has been to an overall advantage during the course of the study, it is also important to appreciate that this, albeit not particularly extensive experience, potentially poses a question of bias. Additionally, as the author has met and to a limited extent also worked with most of those interviewed prior to the sessions, consequently this should also be taken into account for possible bias reasons. The author believes the question of bias has been satisfactorily addressed by recognizing the issue and by designing the interviews and the protocol for how the interviews will be analyzed accordingly. By using the rather well-defined ISO 9126-1 definitions of software quality as a tool in analyzing the transcripts, any risk of bias has been reduced to an acceptable level.

Having gone through all transcripts and categorized the answers, a list was compiled where all the responses of the same software quality were grouped together. To differentiate the responses further, they were also categorized into groups of positive and negative remarks. A “positive” remark is something the respondent was happy with in the system such as a function or feature. For example, when asked if they felt the system was user-friendly, one respondent stated:

“Yes, overall I think it’s simple and basic.”

This was consequently categorized as being a positive remark relating to usability. A “negative” remark is something the respondent is unhappy with. It may be an aspect of the current system that the user doesn’t feel satisfied with or some function or feature they wish the system would have. For example, when asked if they feel the system is fast and responsive, one respondent answered:

“No, I wouldn’t say that. It’s about average I suppose. It takes a lot of mouse clicks to confirm things before you can get something done”

This was categorized as a negative remark relating to efficiency.

In the final step the answers in the list were counted and the results compiled into a table (see Figure 8).

<table>
<thead>
<tr>
<th>SQ</th>
<th>Pos:</th>
<th>Neg:</th>
<th>Pos/neg ratio:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>8</td>
<td>22</td>
<td>2.75</td>
</tr>
<tr>
<td>Reliability</td>
<td>9</td>
<td>13</td>
<td>1.44</td>
</tr>
<tr>
<td>Usability</td>
<td>12</td>
<td>29</td>
<td>2.42</td>
</tr>
<tr>
<td>Efficiency</td>
<td>10</td>
<td>10</td>
<td>1.0</td>
</tr>
<tr>
<td>Maintainability</td>
<td>4</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Portability</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

*Figure 8 – Results of interviewees categorized into quality characteristics*

As the table shows, there are overall negative tendencies in most software quality aspects, with maintainability and portability having the highest relative ratio of positive contra negative remarks. However, functionality and usability have the overall highest number of remarks. Issues relating to these two software qualities are the most prominent in the interviews and since there are substantially more negative
remarks than there are positive, it stands to reason that the current system has several weaknesses, in particular related to these two characteristics. Another important point is that because these two software qualities receive the most attention in the study, they are consequently the two most important ones to those interviewed.

7.2 Method evaluation

Among the previously chosen and described methods for software evaluation, one is to be chosen as the best suited methodology for this particular evaluation problem. To reiterate, VTEC is considering upgrading or changing their intellectual property management system. There are a number of available market alternatives and as a step in finding out which one will potentially suit them best, a methodology that will help the company identify their needs and assist them in choosing a system is required.

At this point, it is important to the expected outcome of the thesis and to the project as a whole to make one thing clear. From the beginning, the goal of this thesis has been to identify a method of evaluation that will help the company, Volvo Technology Corporation, in selecting the software solution best suited for their needs based on their organizational requirements. During the course of this thesis it has become abundantly clear that no “magic box”-method that one can simply feed with a set of alternatives and expect it to deliver the correct answer, exist. Instead, the intention of evaluation methods is to function as a tool for decision makers to explore alternatives and if possible deliver results that can help those in charge to make a so informed decision as possible when the time comes to choose an alternative. Thus, one should not rely solely on the outcome of an evaluation method, but rather regard it as a knowledge resource and a means to acquire greater insight into the problem.

Furthermore, while the outcome of some evaluation methods is a recommendation or ranking of alternatives, some methods serve as an apparatus for the organization to define in greater detail what requirements, functions, features, etc. they want from the type of software product under evaluation. The successful implementation of such a method would ideally make the choice between alternatives a process of matching the outcome of the method with the specifications of the alternatives and thus identifying the best suited candidate.

The organizational study was conducted with the ISO Software quality model as basis for the interviews. The six quality characteristics encompass together with the sub-characteristics defined by Losavio et al. (2003) cover many of the requirements that an organization has on a software product (Losavio et al. 2003). There is however one aspect that the ISO model does not seem to discuss, that being the software providers responsibilities towards the customer. For example, issues such as the providers’ obligation to sustain a level of maintainability in the delivered system are not covered by any of the six software qualities.

Regardless, one could argue that for a software evaluation method to be of practical use it must support the inclusion and important domain information that software quality characteristics offers. In order to reach a conclusion regarding which of the evaluation methods would be best in this case, their capability to support software quality characteristics will be assessed.

The section below presents a short evaluation of each of the methods capability to incorporate software quality characteristics into the process, followed by a comparison and selection of an evaluation method in the next chapter.
7.2.1 AHP

In an introductory description of the Analytical Hierarchy Process, the method is defined as a mathematical approach to a given decision making problem that also incorporates and builds upon subjective input from the personal involved in the decision making process (Ho, 2008). As has been stated earlier in this thesis in the description of AHP, this combination of decision makers’ judgment combined with AHPs use of mathematical constructs in the form of an eigenvector is what makes this methodology so flexible and popular in multiple criteria decision making domains.

In the first stage of AHP the main objective is formulated, for example “Choose a new system”. The criteria for the objective are then defined. In a situation such as the one VTEC is in, where the main objective would be to choose one out of a number of IP management systems, the criteria for such a system could well be the six software quality characteristics. Thus, this method is capable of incorporating quality characteristics in the process if the decision makers decide to do so.

For this evaluation problem, the quality characteristics would be inserted into the AHP hierarchy -tree below the main objective followed by the identified system alternatives, as illustrated in Figure 9.

![AHP hierarchy tree](image)

Figure 9 – AHP hierarchy tree

In the next stage the criteria are synthesized, or weighted against each other, to determine their ranking. At this point, the results from the organizational study could serve as a good basis for the criteria ranking. By examining the ranking of software characteristics that the analysis of the interviews yielded, one could use these in combination with managements input to synthesize the criteria before advancing to the pairwise comparison stage.
7.2.2 MCDA

The Multiple Criteria Decision Aid methodology has several similarities with AHP, which according to Vaidya and Kumar, (2006) is a decision making tool based on the core functionality of MCDA. MCDA itself has expanded into a great many number of derivations of the original MCDA methodology, each aimed at fitting certain types of evaluation problems better than the standard method. It is however the original MCDA method that has been described earlier in this thesis.

In MCDA, the most important step is the selection of attributes for the object under evaluation. These attributes greatly affect the direction and focus of the evaluation and must consequently be carefully selected (Vaidya and Kumar, 2006). The attributes are then broken down until their basic and measurable form is identified.

For the evaluation problem that is the focus of this thesis, quality would certainly be an important attribute. As illustrated in Figure 10, quality is a compound attribute that is then broken down into the six ISO software quality characteristics, which in turn spawn sub-characteristics until basic attributes are defined.

Figure 10 – MCDA attribute hierarchy model

Thus, an MCDA approach to the evaluation problem would support the incorporation of quality characteristics into the process, while also having the capability to support other important factors such as cost.

7.2.3 GQM

The Goal Question Metric approach to an evaluation problem is different from the more traditional decision making methodologies. Here the main focus is on the actual goals set forth by the organization, both for the project in question but also long-term business goals and strategic goals (Basili, 1992). Without a properly defined goal, the methodology will most likely generate less usable results. A series of questions are then associated with the goal/goals in an effort to break down the objective into measurable pieces or metrics that are in turn connected to the questions.
This mode of procedure makes it less straightforward to incorporate the quality characteristics. In order to include the characteristics into the process, they have to be derived as questions from a goal (see Figure 11).

<table>
<thead>
<tr>
<th>GOAL:</th>
<th>Increase overall efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUESTION:</td>
<td>What is the current transaction time when creating a new patent case?</td>
</tr>
</tbody>
</table>
| METRIC: | • Average t-time based on selection $n$  
| | • Subjective rating by managers |
| QUESTION: | What is the current time required to generate monthly economy report? |
| METRIC: | • Set of trial-runs $n$  
| | • Subjective rating by user |

Figure 11 – GQM sample question set

The outcome of a successfully implemented GQM method is not necessarily a recommendation as to which out of a set of $N$ alternatives is the best, although it could be utilized for that purpose also. In fact, alternatives do not even have to be part of the equation. Instead, GQM can be implemented in such a way that its outcome offers is a greater insight into the problem and an opportunity for the organization to further refine their requirements definitions. The data gathered in the data collection stage of GQM should, once compiled and analyzed, provide valuable information that can serve as a basis for further evaluations. As Basili (1992) states, GQM can be used within the context of a more general approach to software quality improvement or evaluation. In such a situation the development of GQM models is a task which will use the business driven goals provided by the corporate management and the environment characteristics provided by the project team as inputs to the process. This flexibility makes GQM an interesting and potentially valuable tool for evaluation purposes. However, for the particular evaluation problem that is the focus of this thesis, it could be argued that GQM is inferior in the respect that it is less equipped to handle direct evaluation and rankings of alternatives than MCDM approaches.

### 7.2.4 PECA

This methodology, being an exceptionally flexible one, will allow for the integration of software quality characteristics in the requirements and criteria definition stage. As requirements must be defined before criteria can be formulated, a series of statements that capture software quality would have to be composed, in the form of “needs” as...
described earlier in this thesis. These requirements would then be refined into actual measurable criteria.

A feature that sets this method apart from the other three, is its emphasis on planning the evaluation, including selecting evaluation team members and creating charters specifying essential facts and stating goals and scope, details that might otherwise be overlooked in many projects.

One of PECAs drawbacks is its complexity and high level of flexibility. The methodology allows for a wide range of different techniques to be applied at the different stages in the process. Deciding what technique to apply where is left up to the evaluators to decide, providing limited guidance on how to choose. Thus, the experience and knowledge of the evaluators is a critical factor for the success of method implementation.
8 Selection of evaluation method

This chapter starts with exemplifying the importance of software qualities in an evaluation of this kind while then proceeding to motivate the choice of method.

8.1 Software qualities compatibility

From an organizational perspective, the results from the interviews, as illustrated in Figure 8, showed that a significant amount of issues with the current system were related to the functionality and usability software qualities. For example, many interviewees had serious concerns about the systems lack of consistency in performing certain tasks. Users described how the system would be either unreliable in performing certain tasks or act unpredictably resulting in inconsistent outcomes to seemingly similar or identical tasks. To emphasize the importance of evaluating software qualities, here are a few examples of SQ related issues that some users have with the system. One user said:

“All the functions are there but they just don’t work satisfactorily. For example, it doesn’t handle annual fees consistently […] when an invoice comes in, it’s suppose to kill that time limit and create a new one but that doesn’t always happen."

Another interviewee, while speaking about the usability of the system, remarked:

“To get the system to generate a report of some kind is very tricky. Even if I get it to produce the report, I have to double check everything to make sure it’s correct. You shouldn’t have to do that and it’s all up to what and how you tell it [the system] to generate which is very hard to get right.”

This sort of inconsistency in the systems behavior has made some users feel they cannot rely on the system performing correctly which in turn has created a need for double checking the systems results and ultimately having a negative effect on efficiency and user satisfaction.

The kinds of issues and problems that the interviews highlighted as a result of the ISO-related questions reiterate the value of a ISO-SQ compatible method, which has been one of the primary considerations while evaluating the methods.

8.2 Choosing a method

Having described the methods and discussed their aptness for including software quality characteristics into the evaluation process, no one method can be singled out solely on the premise of if they’re capable of incorporating software qualities as criteria and attributes. Although the author feels this is a critical factor for the methods success, other important key-features of evaluation methods should be taken into account as well, so as to attain a higher level of quality in the evaluation.

Thus, to reach a conclusion as to which method to choose, one has to look at other aspects and characteristics of the methods in conjunction with their ability to evaluate software quality.
Based on the methods overall functionality and general methodology, as described in this thesis, one can deduce important information that can help in establishing their suitability for the current evaluation problem. Some important characteristics have been compiled in Figure 12. The characteristics compiled in Figure 12 are those that were considered most important and relevant to the evaluation problem explored in this thesis. Figure 13 offers a more comprehensive view as to which methods support which features.

<table>
<thead>
<tr>
<th>Features</th>
<th>AHP</th>
<th>MCDA</th>
<th>GQM</th>
<th>PECA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports SQ in evaluation</td>
<td>✓</td>
<td>✓</td>
<td>◊</td>
<td>✓</td>
</tr>
<tr>
<td>Synthesization and pairwise comp. of attributes</td>
<td>✓</td>
<td>◊</td>
<td>✗</td>
<td>◊</td>
</tr>
<tr>
<td>Ability to rank alternatives</td>
<td>✓</td>
<td>✓</td>
<td>◊</td>
<td>✓</td>
</tr>
<tr>
<td>Can include other org. factors in evaluation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Non-sequential execution of activities</td>
<td>✗</td>
<td>◊</td>
<td>◊</td>
<td>✓</td>
</tr>
<tr>
<td>Supports multiple techniques</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Long-term org. perspective</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>◊</td>
</tr>
<tr>
<td>Well defined structure</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easy to follow / user-friendly</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Figure 12 – Key aspects of evaluation methods

Figure 13 – Features supported by the four methods

✓ = Native support for feature
◊ = Not natively supported
✗ = Feature not supported
Some features are not natively supported by the method (marked by a “◊” in the matrix), for example, GQM does not offer a clear and pre-defined way of incorporating software quality characteristics in the evaluation. However, SQ can be evaluated but requires some degree of modification of the original methodology.

To reach a conclusion on which method to choose, each of the key-features in Figure 13 will now be discussed in further detail:

**Supports SQ in evaluation**

As presented in Chapter 7.2, all methods will support the ISO software qualities in an evaluation. AHP and MCDA do have an advantage compared to GQM and PECA in that they both are able to incorporate software qualities and integrate them better into the evaluation. AHP in particular does this well by making it possible to use the synthesization and pairwise comparison techniques to further take advantage of SQ.

**Synthesization and pairwise comparison of attributes**

As has been explained previously, this is a stage before the actual evaluation of the set of alternatives take place. The attributes and criteria that the evaluation will be based upon is defined and the evaluators have an opportunity to weigh then criteria against each other, thus making it possible to include considerations about the criteria that might have otherwise been left out. AHP is the only out of the four methods that include this stage natively. While it is possible to apply this step in MCDA and PECA, AHP is superior as it is an integral part of the methodology.

**Ability to rank alternatives**

Furthermore, as the objective is to evaluate and rank a set of product alternatives, the methods ability to deliver a recommendation as to which product is “best”, is another factor to consider. Although all four methods could potentially accomplish this, GQM is clearly the weakest in this respect, as it is more aimed at being used in combination with other evaluation methods, to provide better insight into the organizations requirements and goals. AHP is the only of these four method that employ a mathematical approach, using the eigenvector, to rank alternatives based on subjective input from evaluators and stakeholders (Ho, 2008). The combination of subjective input and a mathematical approach gives the method an advantage when the goal is to rank a set of alternatives, in that it provides a logical framework to determine the benefits of each alternative (Saaty, 1980).

**Can include other org. factors in evaluation**

Inclusion of other factors besides software qualities is possible in all four methods. The methods are all flexible enough so that evaluators can incorporate criteria and attributes after what the situation and problem at hand requires. For example, cost considerations for the product alternatives can be included either as part of the main evaluation, or as a separate entity. It is preferable to set costs aside until the benefits of the alternatives are evaluated as these two interests often conflict (Vaidya and Kumar, 2006). In AHP, Cost to Benefit ratios can be calculated using the results from the product evaluation.
Non-sequential execution of activities
This feature is dependent on the methods flexibility and may be important in many evaluations. However, a method such as PECA which is highly flexible and thus support non-sequential executions also requires more in terms of experience and knowledge from the evaluators. The ability to perform non-sequential executions is more important in IS development projects where being able to iterate phases is crucial. In this case, where the focus is on making a choice out of a set of alternatives, a more structured and sequential method is to favor, such as AHP or MCDA.

Supports multiple techniques
Again, this feature is related to the methods flexibility. Most methods will allow for different techniques to be used in the various stages of evaluation. Although the ability to choose between different techniques may be an advantage in some evaluation scenarios, a method that prescribes specific techniques for each step is likely to be more structured and provide more in terms of guidance and support to the evaluators.

Long-term org. perspective
Clearly, GQM is the methodology that supports a long term organizational perspective best out of the four candidates. While such a perspective may be very important to have in some evaluation situations, it can also take focus from the actual evaluation. AHP and MCDA lack an overall consideration to long term perspectives, instead focusing on the current evaluation problem. For this reason both types of methods would benefit from being applied in an evaluation.

Well defined structure
Having a well defined structure means that the methodology has a well documented approach for how the method is to be applied. This is important to the evaluators implementing the method as it gives them something to fall back on during the evaluation process. AHP is well structured in that all steps in the method are clearly defined, making the evaluation easier to conduct while also providing a sort of quality control. If the method structure has been followed it increases the chances that the method has been correctly applied, thus the result is more reliable. This is also related to flexibility and complexity of the method. A more flexible method, such as PECA which provides many alternatives to how each step can be accomplished in combination with the ability to perform them in a non-sequential manner, is less structured. Arguably, this can be an advantage in more complex evaluation projects but requires a lot more from the evaluators.

Easy to follow / user-friendly
Another important factor is the methods dependence on experience from those tasked with performing an evaluation. Obviously, regardless of which method is applied, experience will have an effect on the outcome of an evaluation. Having skilled and knowledgeable people involved in the process is desirable in all projects but GQM and PECA are especially dependent on the experience of its implementers. This is in part due to their flexibility and complexity. PECA’s ability to incorporate many different techniques in for example the data collection stage requires that those involved know which technique to choose. GQM, although quite popular and well documented, provides little guidance on how it
is to be used, instead relying on the experience and competence of the personal involved in the evaluation.

AHP, on the other hand, has a detailed structure and is also very well documented, providing those conducting the evaluation with a relatively straightforward approach.

On the basis of the previous chapters’ method evaluation in combination with the investigation of the methods support for some important key-features, the Analytical Hierarchy Process (AHP) appears to be the most suitable choice for the evaluation problem explored in this thesis.

In summary, the method easily integrates the use of software quality characteristics into the evaluation while also providing many other advantages. The pairwise comparison of the decomposed attributes is an effective way of imposing a quality control in the criteria specification stage. The defining of evaluation criteria is an activity all methods declare as essential to the process, although AHP is the only out of the four that specifically has a function for assuring criteria quality and consistency. However, Comella-Dorda et al. (2004) mention pairwise comparison and the possibility to include it in PECA.

Perhaps the most important factor, and one of the more prominent reasons to opt for AHP, is that the information elicited from the organizational survey can be of great help in the synthesization and pairwise comparison of criteria, which AHP is the only out of the four methods to support. By examining the software quality ranking results presented in Figure 12, the subjective input from the users of the system can be represented in the evaluation, as opposed to only basing the weighting of criteria on the input of those involved in the actual evaluation.

In conclusion, while the AHP methodology is the recommended choice in the scope of this thesis, it is important to reiterate that the other methods may well be better suited than AHP in other situations and contexts. Thus, this conclusion should not be interpreted as a way of saying that AHP is overall superior to any other software evaluation techniques.
9 Conclusion

The results and deliverables of the thesis, along with a final analysis and possible future work, are presented below.

9.1 Summary

The aim of this thesis was to identify and evaluate a set of software evaluation methods in order to ultimately choose one of them which could be of help in choosing a suitable IPMS solution for the patent department of Volvo Technology Corporation (VTEC). To accomplish this, the first step was to explore what evaluation methods exist and which of these were most prominent and applicable to the problem. Four methods were singled out as being the most suitable candidates.

Secondly, an organizational study, in the form of interviews of system users and administrators, was performed with the intention of gaining knowledge and information about the problem, the current systems advantages and shortcomings and to get a general idea of what characteristics an ideal IPMS (according to VTEC) should have. Results from the study were analyzed, mainly based on the ISO 9126-1 Software Quality Model. The previously identified evaluation methods were evaluated based on different parameters, among those to what degree they support the incorporation of software qualities and how well they apply to the organization and the problem at hand. The Analytical Hierarchy Process (AHP) was deemed to be the best choice for evaluating IPMS solutions for VTEC.

9.2 General applicability of evaluation technique

The four software evaluation methods that were chosen for this project are not limited to the particular type of evaluation problem that is the object of this thesis. On the contrary, in the process of choosing which methods to evaluate, the purpose was to identify those that are considered the most widely used and generally acknowledged to be the most prominent software evaluation methodologies, applicable to a wide range of evaluation problems.

In a similar fashion, the process of evaluating and ultimately reaching a conclusion as to which method to choose for the current problem, is an approach the author believes can be applicable to other evaluation problems in other domains and contexts, outside that which is the scope of this thesis.

Perhaps, calling it a methodology is too strong a word at this point. It is however a mode of procedure that has been developed out of necessity under the course of this thesis project, since no established method for evaluating methods was identified and utilized. The approach consists of a number of steps, as illustrated in Figure 14.
Figure 14 – Steps for choosing a method of evaluation

The execution of these steps will need to be tailored to the organization and the specific evaluation problem under consideration. In the second step, in which an initial selection of evaluation techniques is performed, the result of this step in this thesis may be sufficient to adopt, saving both time and resources. Indeed, several other results and conclusions in this thesis could hopefully be of use such as the assessment of the four candidate methods. However, it is important to emphasize that the conclusions reached in this thesis as to which method to opt for is in no way meant to be interpreted as ruling the other three methods inferior or to be of lesser quality. The choice of method is applicable only for the circumstances set forth in this thesis and potentially also to other similar evaluation problems that share characteristics with this one.

9.3 Deliverables

The results of this thesis are of both academic nature as well as more tangible and practical nature which can be of use to VTEC. Actual deliverables that may be of value to the company are:

- **Evaluation method**
  - Recommendation on software evaluation method to use in a future evaluation of IPMS solutions

- **Information from organizational study**
  - Results from interviews reflecting user opinions and experiences with the current system
9.4 Final analysis

The research area this thesis belongs to is one that has been explored, researched, written about and expanded for circa thirty years now (Bernroider and Stix, 2005). Thus, the amount of information that has accumulated over the years is immense and difficult to survey. An important component in assessing the quality of the findings in this thesis, is to consider the relevance of the four evaluation methods chosen. As has already been discussed in chapter 5, considering such factors as time constraint and the knowledge and experience of the author, no guarantees can be made that those four methods are the “correct” or “best” choice. It is entirely possible that a more extensive evaluation of software evaluation methods would differ in its selection of methodologies to study further and, thus, ultimately reach different results than this thesis.

However, given the prerequisites of the thesis, the author feels the research question originally posed has been answered satisfactorily with sufficient logical and scientific reasoning to back up such a claim.

9.5 Future work

As previously discussed, the project aim was developed in cooperation with VTEC. Ultimately, the goal for VTEC is to make a decision as to what IPMS solution to use in the future. This purpose for this thesis was to use its results as a good stepping-off point for possible future evaluations. Additionally, the knowledge gathered during the organizational study could be useful in such an evaluation. Thus, from VTEC’s point-of-view a possible continuation of this project would involve an actual evaluation of software solutions using the prescribed method from this thesis.

From an academic perspective, future work based upon this thesis could well entail further development of a method for evaluating software evaluation methods. The fundamentals for such an approach have been explored and defined in this thesis and the author feels that the development of such a method could be a real contribution to this field of research. In developing this method, it would be interesting to expand it so that it can be applied to other kinds of evaluation projects, widening its applicability and use to other contexts and domains.

The primary reason why this would be a valuable contribution is the vast and continuously growing range of evaluation methods available which makes choosing the right method for the job, a difficult task requiring, among other things, a good deal of knowledge and experience.
References


Appendix

List of acronyms:

AHP  Analytical Hierarchy Process
COTS Commercial-off-the-shelf
GQM  Goal-Question-Metric
IP   Intellectual Property
IPMS Intellectual Property Management System
IS   Information System
ISO  International Organization for Standardization
IT   Information Technology
MCDA Multiple-Criteria Decision Aid
MCDM  Multiple Criteria Decision Making
PECA Plan, Establish, Collect, Analyze
TPS  Transaction Processing Systems