Abstract. In 2001 the first version of MoReq, a requirements specification for records management system (RMS), was made available online. With the introduction and spread of MoReq in Europe the RMS’s consumers started to consider its compliance a demand, leaving a problem for suppliers with products already developed: how to evaluate their software based on MoReq and what developments are necessary to meet its requirements. In this dissertation we propose a process for evaluating content management systems, class of systems where the RMS fall, according to a requirements specification as MoReq. To achieve the final results we studied and applied techniques for evaluating software in an already developed document management system (SmartDocs) who wants to evolve to a system in full compliance with the requirements of MoReq2, the second version of MoReq.

Keywords: document management system; content management system; record management system; MoReq; software evaluation; software evolution;

1 Introduction

A Document Management System (DMS) it’s an information system conceived to manage all the phases of the document lifecycle. Those are typically the capture and creation of the document, its storage, version management, access and eventually elimination. A DMS is now essential to increase the productivity of a company, which is why there are few who do not have one implemented and there are many solutions available in the market.

A DMS is also called a content management system since it manages documents as content. Another type of content management system is a record management system (RMS), an information system capable of managing records of an organization considering its archive’s requirements. A record is according to ISO 15489 “information created, received, and maintained as evidence by an organization or person in the transaction of business, or in the pursuance of legal obligations, regardless of media”.

In 2001 MoReq, a model of requirements for RMS, was published by DLM-Forum (with the support of the European Commission). Despite the fact that it’s not a formal standard, but rather a set of recommendations, CMS users are starting to demand systems in conformance with MoReq. For that reason CMS developers are looking to know how to evaluate their products according to MoReq so that they can learn which kind of developments are needed so that their software can be in conformance.
The goal of this project is then to propose a content management system evaluation process according to a requirements specification like MoReq. The proposed solution is a result of the lessons taken from the use of evaluation methods in a DMS product (SmartDocs) using MoReq2, second version of MoReq, as conformance goal to achieve.

2 State of Art

2.1 Software Evolution

Software evolution, or software maintenance, is according to IEEE Standard 1219 “the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment”. The concept of software evolution started to appear in 1980 with the definition of Lehman’s Evolution Laws and the categorization of maintenance activities by Swanson (Lehman, Programs, Life Cycles, and Laws of Software Evolution, 1980) (Lehman, Perry, & Ramil, Implications of evolution metrics on software maintenance, 1998) (Swanson, 1976). Swanson defined 4 major maintenance activities classes (Adaptive, Perfective, Corrective and Preventative) that are still currently used but with different terms and different definitions (ISO/IEC-14764, 2006):

- **Corrective maintenance** are changes that fix bugs in the codebase.
- **Adaptive maintenance** are changes that allow a system to run within a new technical infrastructure.
- **Perfective maintenance** are any other enhancements intended to make the system better, such as adding new features, boosting performance, or improving system documentation.
- **Preventive maintenance** are changes made to ease future maintenance and evolution of the system, such as re-organizing internal dependencies to improve cohesion and coupling.

However some software evolution researches believe that 4 classes are insufficient to enclose all possibly maintenance activities so their started to create their own classifications. One of the most successful classifications was made by Ned Chaplin (Chapin, Hale, Khan, Ramil, & Tan, 2000). Chapin also created 4 major classes that divided maintenance activities by type of software changes and inside each class the activities are divided by their objectives. Figure 1 taken from (Chapin, Hale, Khan, Ramil, & Tan, 2000) shows the decision tree that can be used to classify a maintenance activity.

As you can see in figure 1, one of the major classes is evolution activities where software isn’t changed entitled support interface. Inside that class Chapin considered that the main activity is the evaluation of software that involve activities as auditing, searching, examining, regression testing, studying, and others activities that can create an understanding of the software without changing it (Chapin, Hale, Khan, Ramil, & Tan, 2000). The activities considered on this project are then evaluative and fit the A-3 group activity of Chapin.
2.2 Software Evaluation

As we saw in last chapter, the software evaluation is a software evolution activity that was identified and classified many years ago. However DMS evaluation is a fresh activity since only in 1997 (with the creation of the DoD 5015.02 standard) best practices for document management where defined.

One of the most known articles about DMS evaluation is the article of Asprey and Middleton entitled “Specifying your requirements and selecting your supplier for records and document management applications” (Asprey & Middleton, 2002) where the authors address two major problems of today: the creation of a requirements specification and the selection of a document management software from a set of available solutions. Is on the second problem that Asprey and Middleton write about the use of evaluations methods where we can use the results to classify a system or taking conclusions about it self. Despite the fact that the goal of this project thesis is not the same of the authors of that article we can use the methods refered to achieve our goal. The techniques that Asprey and Middleton considered where: checklists, weighted scored and gap analysis.

2.2.1 Checklists

Checklists for software evaluation usually consist of a list of items which are structured intuitively according to some major categories of the product. The items
can be questions, requirements and/or functionalities of the system. The major advantages of a checklist are (Tergan, 1998):

- Evaluation checklists provide a structured list of relevant criteria for evaluation and relieve evaluators from either developing an evaluation system of their own or asking different experts to do the job. Sometimes they may replace large scale expensive empirical testing.
- They seem to be easy to handle. Evaluators usually only have to go through the list and mark whether the product matches, or not, the item.
- They seem to induce the impression of a complete set of evaluation criteria valid, objective and reliable through a simple and cheap method.

The major disadvantages are (Komoski, 1987) (McDougall & Squires, 1995):

- The checklist evaluation emphasis similarities rather than differences forgetting about the quality of the product above the list.
- The evaluation is very subjective to the evaluator’s thoughts.
- The correctness and completeness of the list is essential to the quality of the evaluation. Bad or missing items can lead to a bad evaluation.
- The same type of response for all items may not make sense for all lists.

2.2.2 Weighted Scored

Weighted Scored is one of the most used evaluations and is based on checklist evaluations. The method consists of:

1. Define a weight for each item on the list that reflects its importance on the set of items.
2. Define the range of possible responses to each item on the list.
3. Give a response to each item.
4. Calculate the score of the product in evaluation according to this formula:

   \[ S(x) = \sum_{j=1}^{n} W_j S_{ij} \]

   Where \( W_j \) is the weight of the item \( j \) and \( S_{ij} \) is the response \( i \) to the item \( j \).

2.2.3 Gap Analysis

The gap analysis is in its usual practice, a technique for analyzing the differences between the current state of software and that to be achieved. The definition of interval analysis is the space between the state where the software is and where it wants to be. This type of analysis helps to understand the amount of work that needs to be done to evaluate the software and clarify what are the main activities that need to be done between the two states (Levine, A Sample Gap Analysis Explained, 2010) (Levine, Learn About Gap Analysis Methods: Wich is the Best?, 2010).

To correctly establish the bridge between the two spaces is important to understand not only the state you want to achieve, but also the state where we are, i.e. you must understand the system correctly and recognize its strengths so that we can use them to achieve our goal.
3 SmartDocs

SmartDocs is a DMS developed by Fujitsu Portugal that has already 170 installations and more than 20,000 users estimated. The objective of the program is to manage documents in a fast and efficient way so that organizations that use it can have total control over their information.

According to a comparison made against other systems we conclude that SmartDocs has all the major functionalities of a DMS included in the following areas:

- **Classification Scheme.** The program has functions to create and maintain a structure or base to store and identify documents.
- **Identification and Categorization.** The software includes functions to identify and classify a document like metadata, annotations, categorization, etc.
- **Document Capture.** SmartDocs have more than one way possible to capture a document like for example capturing a document through email.
- **Library Services.** The program has functions to improve the management and control of documents like report creation, version control, audit trail, etc.
- **Search.** The software allows simple and advanced searches.
- **Workflow.** SmartDocs has functions to create and manage workflows that improve the flow of a document between users.
- **Security.** The program has a system of access permissions.

4 MoReq

The MoReq became available electronically for the first time in 2001 and was posted officially by the EU Commission in 2002. It is currently available online at Europa site\(^1\) and has been translated into 7 languages. Since it became available, MoReq has proved to be a very useful specification and started to being used in several countries in Europe. However, with the rapid growth and development of information technology some people began to recognize some outdated content and started to demand a MoReq update so that it can maintain its value as a reference model. That’s why in 2008 a second version of MoReq was launched called MoReq2.

4.1 MoReq2 Analysis

MoReq2 currently has 792 requirements divided into 13 chapters and nine annexes, with over 70 categories. The version in Word format contains 377 pages and over 91,000 words. Summarizing the specification is quite extensive and complex which means that it needs to be analyzed for total understanding of their requirements.

The first chapter is introductory and talks a bit about the history, purpose and organization of the specification. The second chapter introduces an overview of a few

concepts of records management. From chapter 3 to 9 MoReq described its main requirements dividing them by subject:

– Chapter 3 (“Classification Scheme and File Organisation”) lists the requirements about the configuration and management of the classification scheme of a RMS and its entities: classes, files, sub-files and volumes.

– Chapter 4 (“Controls and Security”) describes the requirements concerning the access to records and functions, the maintenance of the audit trail and maintenance of backup and recovery schedules. It also introduces the requirements for supporting the concept of vital records.

– Chapter 5 (“Retention and Disposition”) reviews the requirements about the retention and disposition schedules and the support of the records disposition that includes reviewing, transferring, exporting or destroying the record.

– Chapter 6 (“Capturing and Declaring Records”) lists requirements about the capture of records, bulk import, email management, capture through scanning and imaging and support of creating and maintaining record types.

– Chapter 7 (“Referencing”) describes the requirements concerning the classification codes of the entities of the system and also theirs system identifiers.

– Chapter 8 (“Searching, Retrieval and Presentation”) like the name says lists requirements about the search of records, its retrieval and presentation.

– Chapter 9 (“Administrative Functions”) describes the requirements concerning general administration of the system like the access to system reports and configuration parameters of the system. It also describes the concept of redacting records through some of the requirements in this chapter.

Chapter 10, 11 e 12 are respectively requirements for optional modules, non-function requirements and metadata requirements, that where studied but not used in this project. The last chapter (13) is a summary and revisitation of the concepts used in the specification.
5 SmartDocs Evaluation according to MoReq2

After analyzing SmartDocs and MoReq2 we applied the evaluation techniques described in 2.2. and studied the results.

5.1 Checklist Evaluation

For this evaluation we used MoReq2 has a list and its requirements as the items of the list. So the process evaluation has to go through the requirements of the specification and indicate if it is fulfilled by SmartDocs. Table 1 shows the final result of the evaluation where the last column indicates the percentage of requirements through the chapters and sub-chapters of MoReq2.

<table>
<thead>
<tr>
<th>Requisitos Obrigatórios</th>
<th>Número de Requisitos</th>
<th>Requisitos Cumpridos</th>
<th>Requisitos Cumpridos (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plano de Classificação e Organização de Pastas</td>
<td>72</td>
<td>21</td>
<td>29%</td>
</tr>
<tr>
<td>Configuração do Plano de Classificação</td>
<td>19</td>
<td>5</td>
<td>26%</td>
</tr>
<tr>
<td>Classes e Pastas</td>
<td>12</td>
<td>6</td>
<td>50%</td>
</tr>
<tr>
<td>Volumes e Sub-Pastas</td>
<td>18</td>
<td>5</td>
<td>28%</td>
</tr>
<tr>
<td>Gestão do Plano de Classificação</td>
<td>22</td>
<td>5</td>
<td>23%</td>
</tr>
<tr>
<td>Segurança e Controle</td>
<td>45</td>
<td>22</td>
<td>49%</td>
</tr>
<tr>
<td>Acesso</td>
<td>21</td>
<td>13</td>
<td>62%</td>
</tr>
<tr>
<td>Rotina de Auditoria</td>
<td>15</td>
<td>8</td>
<td>53%</td>
</tr>
<tr>
<td>Salvaguarda e Recuperação</td>
<td>5</td>
<td>1</td>
<td>20%</td>
</tr>
<tr>
<td>Documentos de Arquivo Vitais</td>
<td>4</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Detenção e Destino</td>
<td>31</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Tabelas de Seleção</td>
<td>34</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Visão do Destino</td>
<td>7</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Transferência, Exportação e Eliminação</td>
<td>20</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Captura e Declaração de Documentos de Arquivo</td>
<td>53</td>
<td>22</td>
<td>42%</td>
</tr>
<tr>
<td>Captura</td>
<td>26</td>
<td>13</td>
<td>50%</td>
</tr>
<tr>
<td>Importação em Bloco</td>
<td>8</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Gestão de Correio Eletronico</td>
<td>15</td>
<td>5</td>
<td>33%</td>
</tr>
<tr>
<td>Tipos de Documentos de Arquivo</td>
<td>5</td>
<td>5</td>
<td>100%</td>
</tr>
<tr>
<td>Imagens e Digitalização</td>
<td>9</td>
<td>6</td>
<td>67%</td>
</tr>
<tr>
<td>Referenciación</td>
<td>12</td>
<td>9</td>
<td>75%</td>
</tr>
<tr>
<td>Códigos de Classificação</td>
<td>8</td>
<td>5</td>
<td>63%</td>
</tr>
<tr>
<td>Identificadores de Sistema</td>
<td>4</td>
<td>4</td>
<td>100%</td>
</tr>
<tr>
<td>Pesquisa, Recuperação e Visualização</td>
<td>34</td>
<td>17</td>
<td>50%</td>
</tr>
<tr>
<td>Pesquisa e Recuperação</td>
<td>19</td>
<td>11</td>
<td>58%</td>
</tr>
<tr>
<td>Apresentação: Visualização de Documentos de Arquivo</td>
<td>1</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>Apresentação: Impressão</td>
<td>13</td>
<td>5</td>
<td>38%</td>
</tr>
<tr>
<td>Apresentação: Outros</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Funções Administrativas</td>
<td>2</td>
<td>1</td>
<td>50%</td>
</tr>
<tr>
<td>Administração Geral</td>
<td>3</td>
<td>2</td>
<td>67%</td>
</tr>
<tr>
<td>Relatorios</td>
<td>17</td>
<td>4</td>
<td>24%</td>
</tr>
<tr>
<td>Alterar, Eliminar e Truncar Documentos de Arquivo</td>
<td>16</td>
<td>7</td>
<td>44%</td>
</tr>
</tbody>
</table>

Table 1. Table of checklist evaluation results of mandatory requirements in MoReq2.
As we can see the results of the evaluation indicates that SmartDocs in almost all chapters of MoReqs fails to comply with its requirements. However after studying the list and the results we conclude that:

- A lot of requirements are not fulfilled by small details or are only partial fulfilled.
- SmartDocs have almost all the functionalities identified in MoReq2 but the results don’t reflect that.

5.2 Weighted Scored

Taking in account the conclusions made from the last evaluation we made some changes before applying the weighted scored method:

- A restructuring of MoReq2 was made identifying requirements that describes a functionality of the system and requirements that describe a characteristic or attribute of a functionality. According to that the first ones where entitled “major requirements” and the seconds ones “sub-requirements”, the one that are not included in neither one of the categories described are entitled “normal requirements”
- According to the change above we define the weights of which requirement of MoReq2 like is illustrated in table 2.

<table>
<thead>
<tr>
<th>The final weight is the result of multiplying both measures.</th>
<th>Optional Requirement (Weight 1)</th>
<th>Mandatory Requirement (Weight 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major Requirement (Weight 4)</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Normal Requirements (Weight 3)</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Sub-Requirements of a mandatory major requirement (Weight 2)</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Sub-Requirements of an optional major requirement (Weight 1)</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 2.** Table of weights of MoReq2 requirements.

- Instead of only indicating if the requirement is fulfilled or not in this evaluation we used the responses defined in table 3.

<table>
<thead>
<tr>
<th>Response</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>“No Compliance” – The system don’t complies with the requirement.</td>
</tr>
<tr>
<td>1</td>
<td>“Partial Compliance” – The system partial complies with the requirement but still needs some development to achieve full compliance.</td>
</tr>
</tbody>
</table>
Almost Complies” – The system almost complies with the requirement failing only by a small detail.

“Full Compliance” – The system complies with the requirement.

Table 3. Table of responses used in the weighted score evaluation.

Using the weights and responses above we apply the weighted score evaluation and obtain from the same system (without any change) the results in table 4.

Table 4. Table of weighted scored evaluation results of mandatory requirements in MoReq2

As we can see comparing the results of table 1 and 4 we obtained from the same system much better results. This results proves that the conclusions made in the last evaluation where correct and the changes made to prevent the problems where
successful. SmartDocs still have a few areas that we can see it needs to be improved but most of them are almost in compliance with MoReq2.

### 5.3 Gap Analysis

Of the techniques studied to evaluate a product, the gap analysis is the less rigorous and formal. There are many templates online that illustrate and help to make this type of evaluation, but all have something in common: they are built specifically for each project. Only the aim of evaluation is clear: find out where the software is and where it wants to be. With the results of previous evaluations there is no need to explore again where the software is, even more when there is no new formal method for doing so, leaving us only with the task of realizing the development needed to achieve the ultimate objective: to align the SmartDocs with MoReq2.

So for this evaluation we focus on the requirements that SmartDocs don’t comply and make a list of developments to achieve full compliance with MoReq2. For each development we specify:

- **Title of development**, that indicates the functionality or activity that the development covers.
- **Related Requirements.** Set of MoReq2 requirements that are related to the proposal development.
- **Related Developments,** since some of them are related to the same functionality or area of action.
- **Development Weight,** sum of the related requirements of the development.
- **Development Classification,** according to Swanson classification described in chapter 2.1.
- **Development Description,** summary of all requirements related to the development.

After getting the final list of developments we ranked the developments according to the development weights so that Fujistu can understand the most important and urgent developments to do in SmartDocs.

### 6 Result Validation

After utilizing the evaluations techniques studied we can now take some general lessons from the work done.

#### 6.1 Content Management System Evaluation according to a requirements specification

Considering all 3 evaluation methods and the results that which one produced we can take the following conclusions:

1. All evaluations require high knowledge of both specification and system under assessment. The greater the knowledge of both, the faster and correct the evaluator decisions becomes.
2. Both the evaluation checklist and weighted list are methods that take a long time (average of 80 hours each) to a specification as extensive as the MoReq2, but the second allows us to get a good deal of conclusive data that the first does not allow.

3. The analysis and understanding of MoReq2 are vital to any of the evaluations. The correction of the assessment results by weighted list is directly proportional to the correction of the weights of the requirements, ie, depends on the particular analysis of the specification.

4. The document produced in the gap analysis (list of developments) is what has more value to a team of software development, however, would be a much more complex and time consuming resource to achieve if we didn’t have made the previous evaluations.

Taking in account the conclusions above we propose the evaluation process illustrated in figure 2 with the following and consecutive phases:

1. **Data analysis**, to the CMS and the specification of requirements that are essential, respectively, for the proper evaluation of the system and prioritization of requirements in assessment.

2. **Evaluation**. Phase where we evaluate the CMS and obtain the results to analyze.

3. **Planning**, phase where we studied the results of the previous phase with the goal of planning the software evolution.

---

**Figure 2.** Diagram that defines a process to evaluate a CMS based on a requirements specification.
7 Conclusion

The final objective of this dissertation was to understand the state of art of software evaluation in order to propose an evaluation process to a CMS base on a requirements specification. The area of software evaluation, as outlined in chapter 2, is quite extensive and old. However most of the articles about software evaluation still focus only on the importance of evaluating software as a critical stage in the process of development ignoring its relevance in the process of software evolution. The articles found also rarely refer to requirement analysis as a mean to evaluate, often giving the impression that the management and implements of requirements is solely used in software development and not in its evolution.

These facts were the biggest problem in developing this project but are also the ones who give more value to the results obtained. Knowing that there is little research for this specific problem and that there are business areas in which this knowledge can be applied, the solution has value, at least, as a starting point. The project of (André Veiga, 2010) in which i worked during this project is a clear example of the various areas of business (development of software management, requirements specification and selection of software) that the solution presented here can be used.

8 Bibliography


Veiga, A. ([A submeter no IST em Outubro de 2010]). *Arquitectura de Sistemas de Informação de Referência para Gestão Documental na Administração Pública*. 