UDC 004.413.5:005
Original Scientific Paper

Evaluation Framework for Quality Management Software

Nadica Hrgarek *nadica.hrgarek@gmail.com*
Fürstenweg 144 C Top 40, 6020 Innsbruck

Abstract

Identifying and specifying user requirements is an integral part of information systems design and is critical for the project success. More than 50% of the reasons for the project failure presented in the CHAOS report [36] and study of a US Air Force project by Sheldon et al. [33] are related to requirements. The goal of this paper is to assess the relevant user and software requirements which are the basis for an electronic quality management system selection in medical device companies. This paper describes the structured evaluation and selection process of different quality management software tools that shall support business processes. The purpose of this paper is to help the small to medium size medical device companies to choose the right quality management software which meets the company's business needs.
Keywords: quality management system, medical device, software evaluation, user requirement, weighted scoring model

1. Introduction

Medical device companies have a variety of processes and systems for entering, tracking and managing their quality and compliance requirements. Business processes are often disconnected and not integrated or connected to external computer systems. Processes are identified and documented in the quality documentation (e.g. quality manual, standard operating procedures, work instructions, etc.). Very often, the current process is a manual data entry, paper intensive and error-prone process. Such process has no way to automatically generate necessary escalations based on a minimum threshold of events, and there is no easy way to generate reports for corporate-wide visibility. Having one integrated quality management electronic system can greatly reduce costs associated with compliance and quality, and can improve organization's overall operational efficiency.

A quality management software tool shall support the core activities of the quality management function and efficiently support the processing of context activities. Besides the initial cost, it will take significant time and effort to configure and implement a new quality management software tool. The purpose of this paper is to help the small and medium-sized medical device enterprises (SME) choose a quality management software system.

Identifying and specifying user requirements is an integral part of information systems design and is critical for the project success. It is very common for organizations to pay insufficient or inadequate attention to getting their business requirements first and then to choose the right software which meets the company's business needs.

According to the CHAOS research report [36], only 16,2% of IT projects in the US were successful, while 52,7% were challenged (i.e. completed and operational but over budget, over the time estimate and offers fewer features than originally specified) and 31,1% were impaired (i.e. cancelled at some point during the development cycle). More than 50% of the major factors presented in the CHAOS report that cause software projects to fail are related to requirements. Incomplete requirements and lack of user involvement are ranked at the top of the list (Figure 1). The 2000 updated CHAOS research report [35], entitled "Extreme Chaos", found 23% projects failed, 28 % succeeded and 49% challenged.

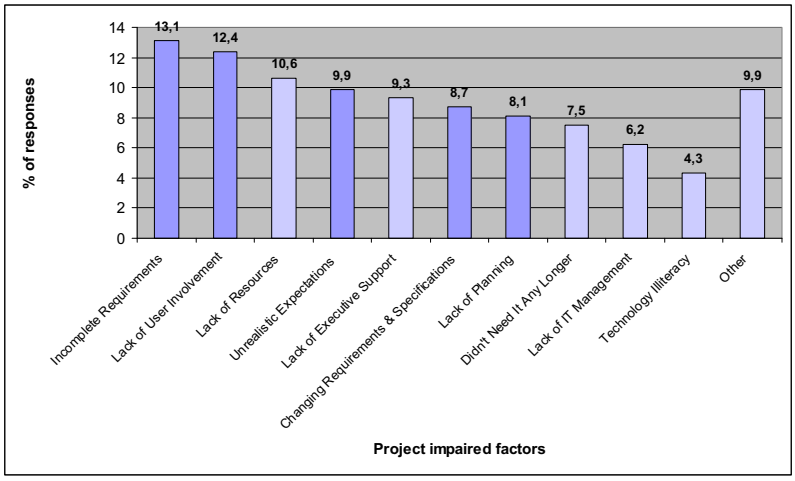


Figure 1. Standish Group project impaired factors (Standish Group, 1994)

In a study of a US Air Force project by Sheldon et al. [33], defects were classified by problem type (Figure 2) and by the phase in the lifecycle in which they are corrected. It was found that requirement defects (requirements translation and incomplete requirements) comprised 41% of the defects discovered, while logic design defects made up 28% of the total error count. Because the costs of problems rise exponentially over the life of an IT project, it is much more expensive to pay for incomplete requirements analysis in the later project phases.

Boehm and Papaccio [6] state that software costs are big and growing – rework costs are smaller in the earlier phases of the software life cycle. This paper provides guidelines on how to identify the user requirements and select appropriate software tool for quality and compliance management.

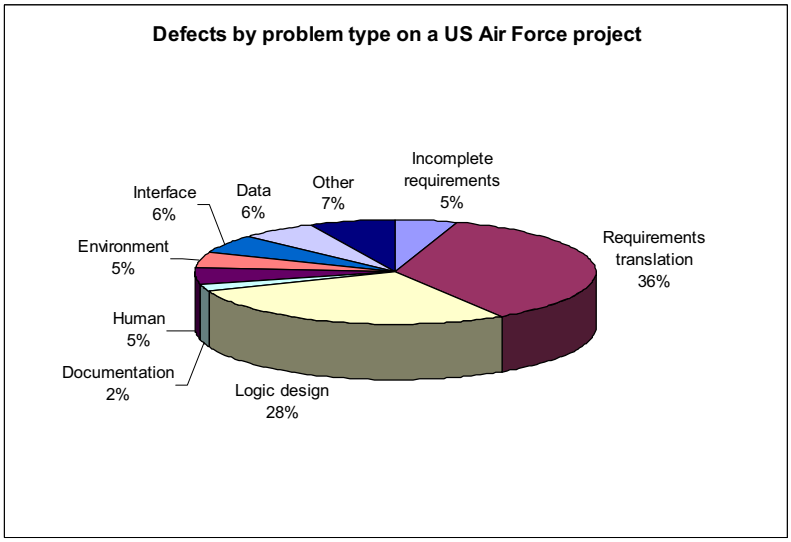


Figure 2. Defects by problem type on a US Air Force project (Sheldon et al., 1992)

2. Quality Management Software Evaluation Criteria

2.1. Requirements

A necessary starting point when selecting a new software system is defining a business requirement list (i.e. evaluation criteria) of what is needed to cover the process workflow (Table 1). In accordance to Kontio [20] the high-level requirements for the software should be

decomposed into hierarchical criteria set and each branch in this hierarchy ends in an evaluation attribute. The evaluation criteria set is specific to each quality management software evaluation case but should include functional requirements, quality attributes, business concerns and constraints. According to Alves et al. [2] the evaluation criteria should include at least functional requirements.

The definitions of a requirement according to IEEE 610.12-1990 [14] are: (1) a condition or capability needed by a user to solve a problem or achieve an objective, (2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents, (3) a documented representation of a condition or capability as in (1) or (2). Requirements types are: business requirements, user requirements, functional and non-functional requirements (e.g. quality attributes, constraints, process requirements).

A user requirement refers to the features or attributes the new system should have or how it should perform from the users' perspective. The user requirement describes what the proposed new system will do, but nothing about how it will be built or implemented. User requirements are narrative descriptions and diagrams showing the services the proposed system will deliver and its limitations.

The good requirements shall be correct, complete, clear and concise, understandable, consistent, traceable, documented, testable, easy to change and implement. Software may be designed, coded, tested and deployed based on an incomplete, incorrect or poor requirement. "Finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase" [5]. If the requirements are inappropriate defined, evaluation of software products is not effective or very difficult to perform.

According to IEEE 610.12-1990 [14] a requirement specification is a document that specifies the requirements for a system or component and typically includes functional requirements, performance requirements, interface requirements, design requirements and development standards. The requirements specification document is the source for all project effort that follows and helps project managers to plan, estimate to complete, staffing needed.

The user requirements specification (URS) is a document which defines what a system should do from a user's perspective. Requirements never specify what has to be done during design or implementation. Users of requirements documents could include various users, internal and external stakeholders (e.g. end users, process owners, managers, requirements analysts, software developers, test engineers and maintenance engineers). The software requirements specification (SRS) and system requirements specification are described in IEEE 830-1998 [15] and IEEE 1233-1998 [13] standards.

2.2. Quality Management Software

A quality management system (QMS) refers to what the company does to ensure that its products and/or services meet the customer's requirements. The quality management system is a set of interrelated business processes (e.g. corrective and preventive actions, change control, document control, risk management, customer complaints management, audit management, equipment management, incident reporting, etc.).

The purpose of quality management software is to track different types of events (e.g. problems, deviations, complaints, audit findings, non-conformances, etc.) into the database and to generate workflows. Workflow means the system can monitor various business processes to detect process errors and notify the user to take action on these items.

A quality management system should include document control, change control, audit process, training of personnel, corrective and preventive actions (CAPA), investigations, maintenance and calibration of equipment, supplier quality management, non-conformance and customer complaints management. All these aspects can be included in a software package to optimally support the quality system. The quality management processes that QMS software supports represent very important selection reason (Figure 3). The quality management software should also offer data for classification, decision making, document

reports and trending. The quality management software tool should adapt to changing regulations and provide quality management that meets the company's business needs, fits company's resources and offers the best price.

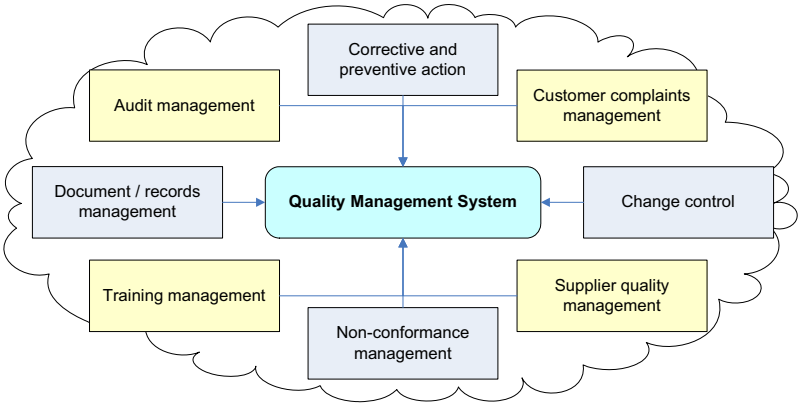


Figure 3. Quality management key processes

With careful evaluation and selection process of the right quality management software, a company may expect to gain significant benefits including increases in quality, productivity and market share, as well as decreases in time and quality problems. The goals to change to a software-based quality management system are focused on, but aren't limited to:

- replace the existing paper-based quality management system and save time due to reduced paperwork,
- automate and improve the quality management processes,
- make work more efficient through the better allocation of resources,
- provide better traceability and accurate information,
- facilitate automated workflows (sequential and parallel) for various users,
- provide programmable notifications and escalations,
- increase accessibility to information (e.g. retrieval using full-text searches),
- reduce time to create, review, approve and publish documents,
- ensure regulatory compliance and reduce the risk of non-compliance,
- minimize the preparation for external audits,
- enhance decision making based on readily accessible data,
- improve customer satisfaction,
- short product release cycle time by closing corrective and preventive actions faster,
- reduce potential for human error due to validated processes.

2.3. Evaluation Criteria

The selection of the best appropriate quality management software solution is a multi-criteria decision problem which can be solved using the weighted sum (also known as weighted scoring) method (WSM), Multi-Criteria Decision Aid (MCDA) methodology [29] or the Analytic Hierarchy Process (AHP) technique [32]. However, there are some limitations of current decision-making techniques [30].

The AHP technique was developed by Saaty in the early 1970's for multiple criteria decision making. The analytic hierarchy process relies on pair-wise comparison between the alternatives to evaluate the importance of the criteria and sub criteria. The AHP is used to prioritize alternatives or determine which alternative best meets a specified goal. This method is supported by software tool Expert Choice and has been successfully applied in software selection [1], [4], [21], [25], [26], [28].

A weighted scoring model is a method that provides a systematic process for selecting tools based on many criteria and weight (Table 1). The evaluation criteria important to the tool selection process have to be identified first and must represent the key areas of

importance. The more detailed and measurable evaluation criteria will result in more reliable evaluation process.

Before constructing an alternatives evaluation matrix (Table 1) a list of alternatives shall be also identified. Then weighting factors between 1 and 5 shall be assigned to each criterion (requirement) to define the level of importance of criteria. Assigning meaning to weighting factors is subjective and may vary from organization to organization (e.g. 5 – very high importance; 4 – high importance; 3 – medium importance; 2 – low importance; 1 – very low importance). The next step is to assign evaluation scores between 0 and 5 to each criterion for each tool (i.e. alternative). Scores are assigned based on subjective judgment. For evaluation scores (fulfillment of a requirement), a typical scale could be: 5 – exceeds requirement; 4, 3 – meets requirement; 2, 1 – partially meets requirement; 0 – does not meet requirement.

Knock out criteria must be satisfied by the software vendor. The not-fulfillment of a single knock out criteria leads to abortion of the evaluation process for the affected tool and vendor is removed from the list. The weighted scores for each criterion are a result of the scoring and weighting factors multiplication. The sum of all weighted scores for each tool gives an indication of the best tool.

			Tool A		**Tool B**		
Requirement group	**Evaluation criteria (i.e. desirable requirements)**	**Weight**	Score	Weighted	Score	Weighted	Knock out
General requirements and pricing	Annually maintenance and support costs	5					
	Centralized administration	4					
	Estimated effort for implementation	5					
	Complexity of administration	4					
	FDA 21 CFR Part 11 compliance	5					x
	FDA 21 CFR Part 820 compliance	5					
	Professional services costs	5					
	Internal IT resources costs (personnel, hardware equipment)	4					
	ISO 13485:2003 compliance	5					x
	Total costs / license fee	5					
	Training costs for IT and end users	5					
Vendor qualifications	Number of employees	3					
	Possibility to audit vendor	3					
	Years in business	4					
	Wide use on the market (number of implementations)	4					
	Financial stability	5					
	Local support	5					
	Medical device reference customers	4					
	Sample plans provided (e.g. implementation plan, training plan)	5					
	Maturity of the product (first release year)	4					
	Willing to demonstrate product at customer site	5					
Security requirements	Audit trails automatically provided for all transaction details	5					
	Electronic signature	5					
	Restricted access to confidential documents	5					
	Role-based access control	5					
	Security at multiple levels	5					
	Password never displays on the screen	5					
Technical requirements	Internet Explorer (web browser)	5					x
	Microsoft® Exchange Server (e-mail server)	5					
	Microsoft® SQL Server (DBMS)	5					x
	Windows Server 2000 (web server)	5					

			Tool A		Tool B		
Requirement group	**Evaluation criteria (i.e. desirable requirements)**	**Weight**	Score	Weighted	Score	Weighted	Knock out
	Windows XP (client)	5					
	Scalability	4					
	Zero software client (no JavaScript, VB Script, Cold Fusion, ActiveX controls, applets)	4					
Versatility	Automated e-mail notifications	5					
	Customization (roles, responsibilities, etc.)	5					
	Dealing with national special symbols, umlaut marks, etc.	4					
	Document linking (cross reference, web links)	4					
	Document numbering (manual, automatic)	5					
	Electronic delivery of FDA applications	5					
	Forms editor / document templates	5					
	Full text search	4					
	GUI workflow editor	4					
	Integration with client e-mail servers (Microsoft® Exchange)	5					
	Integration with other software/applications (e.g. Navision, various measurement systems)	5					
	Keyword search	4					
	Search by attribute (metadata)	4					
	Ability to add and configure new processes (workflow engine included)	5					
	View documents without launching application (through native viewer or other mechanism)	4					
	Workflow management without programming	5					
Business requirements	Additional quality management modules	4					
	Audit management (internal and external audits)	5					
	Calibration management	3					
	Corrective and preventive actions	5					x
	Change control management	5					x
	Clinical trials management	5					
	Customer complaint management	5					x
	Document and records management	5					x
	Equipment (asset) management	3					
	Inspections management	3					
	Maintenance management	3					
	Non-conformance management	5					
	Risk management	3					
	Statistical Process Control (SPC)	2					
	Supplier quality management	5					
	System validation	5					
	Training management	5					
Metrics and reporting	Ability to customize reports	5					
	Built-in reporting or with Crystal Reports	5					x
	Business (performance) metrics	4					
	Charting (different chart types)	5					
	Generating reports in standard file formats (.xls, .pdf, .doc, .rtf, .html)	5					
	Management reports	4					
	PDF publishing	5					

			Tool A		Tool B		
Requirement group	**Evaluation criteria (i.e. desirable requirements)**	**Weight**	Score	Weighted	Score	Weighted	Knock out
	Quality of reports included	4					
	Regulatory reporting (e.g. MedWatch, Vigilance, MedSafe, MHRA)	5					
	Statistical reports	5					
Usability requirements	Consistent and logical navigation flow	4					
	Documentation (e.g. user manual, installation procedure)	5					
	Ease of use – intuitive and simple	5					
	Ease of learning	5					
	Easy on the eyes	5					
	Multilingual GUI	5					
	Response time	4					
	Single point of entry provided for all data input	3					
	Standard GUI features (e.g. pull-down menus, dialog boxes, toolbar buttons)	4					
	Training materials provided	4					
	Web interface	4					
Total							
Final evaluation score in percentage							

Table 1. Weighted evaluation matrix example for quality management tool

3. The Evaluation Framework for Quality Management Software

The use of quality management software in life science and medical device companies is continuously growing. Medical device companies have a lot of processes and are increasingly requesting *configurable* quality management software *systems* which can support and automate an unlimited number of business processes. Beyond *customized systems* are tailored specifically to the customer needs. These systems require custom code to implement user requirements to meet the company's business needs; therefore, they may require more maintenance and validation efforts.

3.1. Related Work

A number of COTS-based selection methods and frameworks have been proposed in the literature [3], [8]. A common approach found in all methods or frameworks described below is the use of AHP and WSM to support the evaluation of COTS (commercial off-the-shelf) software. The Food and Drug Administration (FDA) defines COTS as "configurable, off-the-shelf software" [11], but very often the "C" is understood to mean "commercial". FDA [12] now identifies software only as off-the-shelf (OTS).

The Off-The-Shelf Option (OTSO) method developed by Kontio [20], [21] in 1995 to facilitate a systematic, repeatable and requirements-driven COTS software selection process consists of the following phases: search, screening, evaluation, analysis of results, deployment and assessment. The OTSO method [22] relies on the use of the AHP.

The PORE (*Procurement-Oriented Requirements Engineering*) method for requirements acquisition described by Ncube and Maiden in 1999 [26], [31] supports the evaluation and selection of COTS components. The method is based on the iterative requirements acquisition, evaluation and selection of COTS.

The STACE (*Social-Technical Approach to COTS Evaluation*) framework [23] has been introduced in 1999 [24] to facilitate a social-technical approach to selecting COTS software. This framework comprises four interrelated processes: (1) requirements definition, (2) social-technical criteria definition, (3) alternatives identification and (4) evaluation (assessment).

Comella-Dorda et al. [7] suggest the PECA evaluation process of a COTS product(s) with a recommendation to the decision maker. This process consists of four basic elements: (1) planning the evaluation, (2) establishing the criteria, (3) collecting the data and (4) analyzing the data.

The following standards may also be helpful to identify the requirements and support the software selection process:

- ISO/IEC 9126-1:2001 [19] – defines and gives examples of software quality characteristics (functionality, reliability, usability, efficiency, maintainability, portability) and metrics,
- ISO/IEC 25030:2007 [17] – aim of this standard is to define and evaluate software quality requirements,
- ISO/IEC 25051:2006 [18] – defines quality requirements for COTS software products and requirements for test documentation (test plan, test cases and test results).

3.2. The Evaluation Framework for Quality Management Software

The formal structured tool evaluation and selection process shall be objective and consists of the following four phases:

1. **Requirements acquisition and evaluation planning phase**
   a. **Identification of the problem area, business and functional needs/requirements, not the tool market** – Understanding the business needs and scope (e.g. perform a needs analysis and determine gaps you may have in the current process). Identify processes, determine specific areas for improvement (i.e. today's problems which should be solved with the tool) and develop a high level process map that will capture core business processes. It is important to evaluate these processes and consider whether some processes need to be changed, improved or automated in the future.
   b. **Top management support and project organization** – Defining the tool selection project, budget, project schedule, measurements of project success, an evaluation team and evaluation plan. The effort of selecting and implementing quality management software must have high priority in organization, and management should establish a preliminary budget for the project prior to beginning the project. A preliminary budget shall include software (e.g. application software, database software, document management software and ancillary support software), hardware, implementation, training and project management costs. The project should not be managed by the IT department. The evaluation and selection team shall consist of a project leader (full-time) and some team members (temporary) who know the business needs. Typically, team members are from different functional areas of the company where the tool will be used (e.g. engineering, quality assurance, manufacturing, IT, etc.). Using a project management tool like Microsoft® Project or equivalent for project schedule is recommended. The following milestones should be identified for the software selection project: complete selection, sign software contract, start implementation, train users, go live date and complete implementation.
   c. **Considering the alternatives** – Build versus buy decision shall be reviewed. The criteria considered for the decision could be the requirements, the estimated costs and risks.
   d. **Cost and benefit analysis (CBA)** – Identifying expected costs (e.g. software, new hardware, training, consulting, annual support and maintenance costs, rollout team time) and benefits (e.g. improved business processes leading to an annual cost decrease, better available information and being able to make better decisions, increased employee satisfaction, response to competition).
   e. **Identification and definition of the user requirements and special issues** – The first step of the IT project is specifying the functional requirements, quality attributes and any constraints (e.g. number of users, budget, timescale, regulatory compliance, required or supported hardware and/or software infrastructure, need to interface with

other systems, etc.) for the software system. Functional requirements can be defined with use cases, narrative descriptions, process flow diagrams, etc.

f. **Requirements analysis and review** – It is very important to distinguish mandatory requirements from wish list. Requirements should be reviewed and prioritized. Pursuant to IEEE 610.12-1990 [14] requirements analysis is: (1) the process of studying user needs to arrive at a definition of system, hardware or software requirements, (2) the process of studying and refining system, hardware or software requirements. According to IEEE 610.12-1990 [14] requirements review is a process of meeting during which the requirements for a system, hardware item, or software item are presented to project personnel, managers, users, customers, or other interested parties for comment or approval.

2. **Product identification phase**

g. **Identification what is available on the market** – Defining evaluation criteria. Identifying potential vendors and their software products that could meet the evaluation criteria and should be included for assessment in the evaluation phase. Identification of licensing models and alternatives. Identification of costs which are connected with the ownership of the tool (e.g. license and maintenance fees, training costs, how many IT people are required to run the system, etc.). Identification of limitations and functionalities of the tool. Creating the long list (10-15 solutions). Several sources can be used to identify products available in the market including personal experience, Internet, magazines, product white papers, vendor evaluation studies, exploring how the customer's competitors implemented similar system, articles and other sources of available information.

h. **Identification of tools for a detailed evaluation** – Creating a request for information (RFI), distributing it to potential vendors and requesting that they submit information. Analysis of received information from vendors and creating a weighted evaluation matrix to quickly reduce the list of vendors. The AHP technique shall be used to weigh customer requirements when sensitivity analysis is necessary or to support decisions about critical tool features.

3. **Product evaluation phase**

i. **Execution of a detailed evaluation** – Evaluation of the short listed candidate tools (2-3 solutions). Prepare request for proposal (RFP), review and compare complete vendor RFP responses. Evaluation of the vendor (including documentation, training, support and commercial aspects). Schedule initial remote and on-site product demonstrations by software vendor to see the software in action, not a PowerPoint presentation of software (the selection team must attend all product demos). Total cost of ownership (TCO) comparison. Negotiating with vendors.

j. **On-site demonstration of selected competitive tools** – Perform a proof of concept in a small-scale pilot project (i.e. prototype). Prototypes are useful to observe areas of high technical risk (e.g. interfaces and integration with other systems), user interactions and critical quality attributes (e.g. performance, reliability). Learn more detail about the tool and gain insight how easy is to configure and administer the system. See how the tool would fit with existing business operations and practices, and how they would need to change. Assess whether the benefits will be achieved at reasonable cost. Go through implementation planning with software vendors. All participants shall have the chance to work hands-on with the software in order to evaluate usability and review manuals, help and user guides. Check vendor references, visit client sites, make reference calls, attend user group meetings or conferences.

4. **Product selection phase**

k. **Defining the contract requirements** – Length of contract, licensed needed and expected growth rate, licensing restrictions, integration needs with existing systems (e.g. ERP, MES, LIMS, HRM, CRM, SPC, PLM), deployment costs, updates, training, management requirements, technical requirements, warranties, contract

termination, acceptance criteria, obligations, payment terms and other important issues.

l. **Performing the gap analysis** – It is recommended to perform a gap analysis to determine the gaps between the needs of the organization (i.e. user requirements) and software possibilities. Gaps need to be prioritized. Software vendor should propose solutions to close these gaps.

m. **Making the final decision and contract signing** – Present evaluation results (e.g. evaluation summary with preferred product for recommendation or alternate solutions and rationale for the choices made during the evaluation) to the decision maker (e.g. executive management). Selecting a vendor. If necessary, schedule and perform an on-site visit/supplier audit [27] before signing a contract. Final negotiating and awarding the contract (e.g. vendor support and commitment, implementation support, license model and options, pricing options and flexibility about pricing, etc.). Before signed, the contract shall be reviewed by legal counsel.

n. **Follow up with vendors** – If requested, provide a feedback to vendors (e.g. strengths and weaknesses).

Key success factors for the deployment of the selected tool within an organization include:

- Top management support and involvement.
- Choosing the right software tool that best meets the business requirements.
- Assembled project team, assigned a capable project manager and system administrator.
- Rolling out the tool to the rest of the organization incrementally – Phase approach to implement quality management processes.
- Adapting and improving business processes to fit with the use of the tool.
- Providing training for administrator and new users.
- Defining usage guidelines.
- Implementing a way to learn lessons from tool use.
- Monitoring and measuring tool use and benefits.
- Good project communication.

4. Case Study Results

The case study will not contain any company confidential information including name (hereinafter referred to as "Company A"), address, URL, revenue, number of employees, etc. The purpose of this study was to make the final recommendation on the quality management software tool that would best meet the needs of Company A. It is a very well known company with a reputation for the research and innovative technology in the implantable medical device industry.

The Company A quality management system for medical products is applied and certified according to ISO 13485:2003 [16]. Company A products are required to meet FDA regulatory requirements too. Company A initial goal was to implement a standard, single solution for quality and compliance management across the enterprise. The company wished to provide a fully integrated, user friendly and easy to administer, configurable, enterprise wide quality management information system to support its quality management and regulatory needs. The quality management software should support the Microsoft® SQL Server database management system (DBMS).

4.1. Requirements Acquisition and Evaluation Planning Phase

The user requirements for an electronic quality management system that are identified in this paper are based on a detailed analysis of the business processes operating at Company A and the following FDA regulations and ISO standards: FDA 21 CFR Part 11 [9], FDA 21 CFR Part 820 [10] and ISO 13485:2003 [16].

The requirements which should be taken into account when selecting a system were defined by regulatory affairs, quality assurance and software validation staff. Depending on importance or possibility to be implemented within the project time frame, the requirements were classified as *optional* or *mandatory*.

The requirements were collected using the following requirement elicitation techniques: interviews, meetings with process owners, review of documents (e.g. quality manual, procedures, organization charts, standards, etc.), literature review and brainstorming. Process steps were displayed as flowcharts and integrated in the user requirements specification.

Requirement types identified and defined in the user requirements specification for a quality management software tool were:

- **User functional requirements** – Corrective and preventive actions, document management, change control, non-conformance management, audit management, customer complaints management, training management, supplier quality management, clinical trials management, additional quality management modules (e.g. SPC, failure mode and effects analysis), workflow management without programming knowledge, automated e-mail notifications, reminders and alerts, administration, electronic submissions of FDA applications, formal reporting requirements.
- **Internal/external interface requirements** – User interfaces, hardware interfaces, software interfaces, communication interfaces.
- **Non-functional requirements** – Internationalization and language requirements, performance requirements, data volumes, reliability, availability, stability, ease of use, usability and user interface requirements, user documentation, formal reporting requirements, training requirements, maintainability.
- **Regulatory requirements** – Security requirements, 21 CFR Part 11 electronic records; electronic signatures (data input, data retrieval, data retention, system validation (IQ – installation qualification, OQ – operational qualification, PQ – performance qualification)), access control and password security requirements.
- **Other requirements and system attributes** – Data requirements, records retention, supportability, environment, search, dashboard, charts and graphs, delivery requirements, database support, safety requirements, and future enhancements.

4.2. Product Identification Phase

The structured evaluation of available quality management software tools was initiated in December 2006 and the final selection decision was made approximately six months later. The user requirement specification was used as the basis for the evaluation and implementation of the system. During preliminary online Internet research and marketing material review (e.g. articles, web seminars, technical and white papers, marketing brochures, release notes, vendor references, vendor evaluation studies [34], etc.) twelve potential software solutions that best match Company A business needs were selected and considered for the further evaluation.

Software solutions considered in the initial product identification phase were:

- **CATSWeb®** (AssurX, Inc.) – Web-based enterprise quality and compliance managed solution designed to help companies to track issues and actions.
- **MasterControl™** suite (MasterControl Inc.) – Quality management software (includes electronic document control, training control, CAPA, etc.).
- **Quality*Stream** (MetricStream, Inc.) – Web-based product which is targeted at the quality process management market and enables an enterprise-wide visibility and control of quality management processes (e.g. supplier performance management, quality issue tracking, audits, investigations, complaint management).
- **TrackWise®** (Sparta Systems, Inc.) – Event driven, workflow-based, enterprise quality process management software tool for quality management, compliance management, change control, CAPA, etc. One system to collect, process and expose all quality data into information to improve efficiencies, visibility and decrease regulatory exposure.

- **SmartSolve® Suite** (Pilgrim Software, Inc.) – Comprehensive enterprise compliance quality process management (ECQM), module-based and enterprise-wide application for managing corrective and preventive actions (problem tracking and resolution), complaint handling, internal and external audits, change control, supplier quality management, employee certification and training, and equipment management. The company is focused purely on life sciences markets.
- **Agile Advantage 2006** suite (Agile Software Corporation) – Product lifecycle management (PLM) solution for small and medium enterprise customers.
- **Medical Device Suite™** (Camstar Systems, Inc.) – Enterprise manufacturing execution (MES) and quality management system created specially for medical device manufacturers.
- **EtQ Reliance™ for Medical Device** (EtQ Inc.) – Integrated quality and compliance management system that has been preconfigured to specifically address the needs of the medical device industry.
- **iTAC.MES.Suite** (iTAC Software AG) – MES standardized software solution with core competency in the area of closed-loop traceability. Use of the iTAC.MES.Suite is particularly strong in the automotive and electronics industries as well as in the production of medical devices.
- **CAQ=QSYS®** (IBS AG) – Integrated quality and compliance management solution that helps organizations achieve sustained regulatory compliance and total quality management (TQM).
- **windream** (**win**dream GmbH) – The full integration of a document management system with the Microsoft® Windows operating system.
- **EnterpriseIQ™ Quality Management System** (IQMS) – Complete and comprehensive software package that includes the following features: advanced product quality planning (APQP), CAPA, performance qualification, document control, non-conformance management, calibration management, quick inspections, statistical process control (SPC), deviation tracking, automatic workflow, etc.

4.3. Product Evaluation Phase

During the first part of the structured evaluation process, six companies presented their software solutions by providing installed demo versions and a high level evaluation check list was developed (Figure 4).

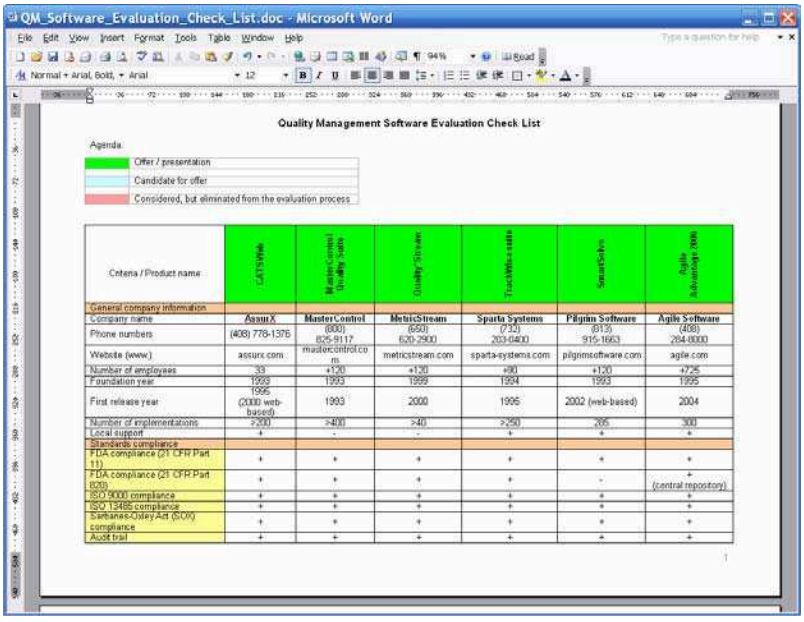
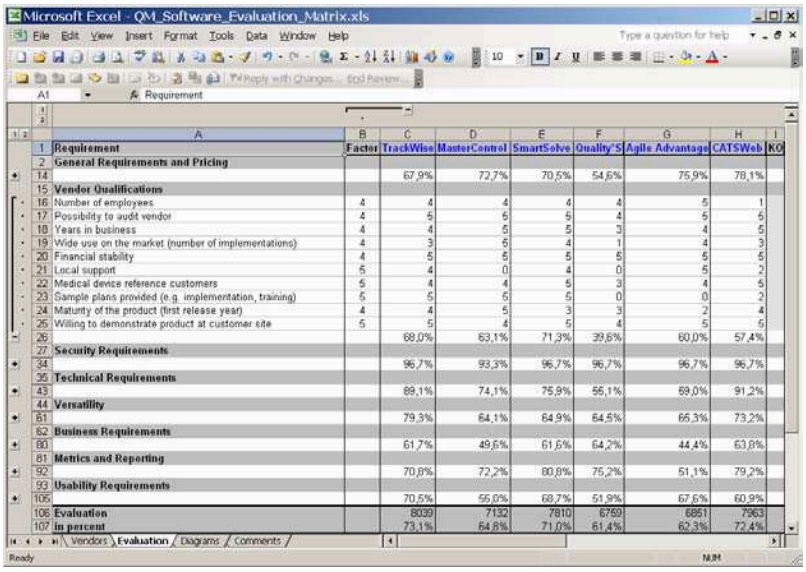


Figure 4. Sample quality management tools evaluation check list

To evaluate the six vendors an evaluation check list and weighted score model (Figure 5) were used. Cost comparison was performed on the same number of concurrent users and professional service days basis.

The comparison matrix produced on a spreadsheet (Figure 5) or as a Word document provided information which software and vendor should be short-listed and evaluated further. This matrix can be used as the basis for more detailed evaluation criteria, for example when attending the software demonstrations. During the evaluation process, it became clear that MetricStream, Inc. and Agile Software Corporation could not meet all Company A requirements. These two companies were unable to provide support for the Microsoft® SQL Server database.

The weighted score model (Figure 5) considered 88 evaluation criteria across eight categories (see Table 1 and Figure 7). The criteria for evaluating alternatives were based on eight key requirement metrics (Table 1). These eight criteria types serve different purposes and include: general requirements and pricing, vendor qualifications, security requirements, technical requirements, versatility, metrics and reporting, business requirements and usability requirements.



	Requirement	Factor	TrackWise	MasterControl	SmartSolve	Quality*S	Agile Advantage	CATSWeb	KO
2	General Requirements and Pricing								
14			67,9%	72,7%	70,5%	54,6%	75,9%	78,1%	
15	Vendor Qualifications								
16	Number of employees	4	4	4	4	4	5	1	
17	Possibility to audit vendor	4	5	5	5	4	5	5	
18	Years in business	4	4	5	5	3	4	5	
19	Wide use on the market (number of implementations)	4	3	5	4	1	4	3	
20	Financial stability	4	5	5	5	5	5	5	
21	Local support	5	4	0	4	0	5	2	
22	Medical device reference customers	5	4	4	5	3	4	5	
23	Sample plans provided (e.g. implementation, training)	5	5	5	5	0	0	2	
24	Maturity of the product (first release year)	4	4	5	3	3	2	4	
25	Willing to demonstrate product at customer site	5	5	4	5	4	5	5	
26			68,0%	63,1%	71,3%	39,6%	60,0%	57,4%	
27	Security Requirements								
34			96,7%	93,3%	96,7%	96,7%	96,7%	96,7%	
35	Technical Requirements								
43			88,1%	74,1%	75,9%	56,1%	69,0%	91,2%	
44	Versatility								
61			79,3%	64,1%	64,9%	64,5%	66,3%	73,2%	
62	Business Requirements								
80			61,7%	49,6%	61,6%	64,2%	44,4%	63,8%	
81	Metrics and Reporting								
92			70,8%	72,2%	80,8%	75,2%	51,1%	79,2%	
93	Usability Requirements								
105			70,5%	55,0%	68,7%	51,9%	67,6%	60,9%	
106	Evaluation		8039	7132	7810	6759	6861	7963	
107	in percent		73,1%	64,8%	71,0%	61,4%	62,3%	72,4%	

Figure 5. Excerpt from the vendor comparison Excel spreadsheet

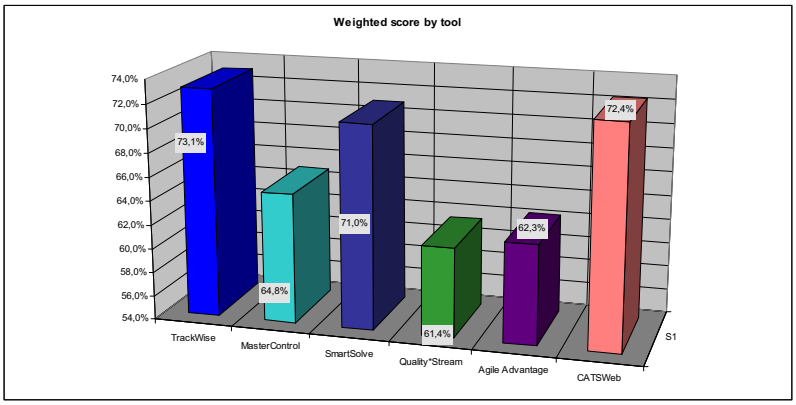


Figure 6. Software tools ranked by weighted score

After checking marketing data and first six on-site and/or remote vendor presentations were conducted, on the basis of selection criteria a number of advantages and disadvantages of implementing a particular software package became clear (Table 2).

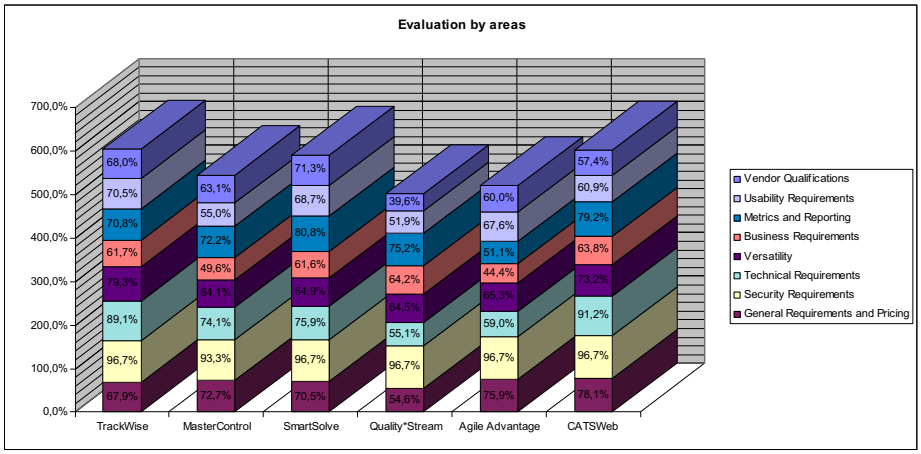


Figure 7. Evaluation of software tools by requirement categories

Software tool	Strengths	Weaknesses
TrackWise®	First commercially off-the-shelf 21 CFR Part 11 solution in the life science industry. Workflow automation (fully configurable without changes to source code or database structure). Configurable software solution that can build an unlimited number of business process and tracking workflows. Tracks all types of events including non-conformance issues, defects, audit results, findings and action items in general. Average implementation time within 4-6 months (40-60 professional service days on-site). A lot of customers in medical device and pharmaceutical industry. Strong business process management engine for complex workflow management. Computer based training (CBT).	Integration with the electronic document management system (DMS) Documentum (light version). To use Documentum additional licenses are needed. For every concurrent license customer gets 10 named users (login accounts). Configuration of the system requires a very detailed analysis and planning of business processes.
CATSWeb®	Ability to add an unlimited number of new forms / workflows. Minimal client hardware requirements. Central repository of data. Modifiable without changes in the core program. Unlimited number of named users. Possible integration with the eDHR. Validation (IQ, OQ) templates. Quickly installation (2 days) – flexibility and ease of deployment. Field sensitive help. KPIs (business metrics). Drill-down capability. Configurable dashboards. Supports all web browsers (e.g. IE, Safari, Opera, Netscape, Firefox). True zero client developed with the ASP and ASP.NET technology.	GUI isn't state of the art (no rich GUI). Some DMS functions aren't supported (e.g. check-in, check-out, exclusive check-out, carbon copy check-out). Configuration requires some programming knowledge.
SmartSolve®	95% complete solution that only requires minor adaptations before production use. Streamline regulatory submissions (FDA, EU, Canada, Japan, and Australia). DMS included. One-click drill down access to data. Designed for multiple organizations to collaborate under one system. Unlimited named users. Extensive technical and user documentation. PDA integration. Calibration certificate, bar code label (equipment). Report ability analysis through decision trees and many predefined reports. Preconfigured templates and workflows. Most comprehensive validation packs. Time-out feature. Discussion forum.	No new forms can be created (customizable and extensible existing forms). Additional modules are available at additional costs. Documents stored in the database can be accessed only from the system. GUI isn't very intuitive.
MasterControl™	Additionally Electronic Device History Record (eDHR) software. Quickly implementation (within 2 weeks). Certification, validation and training support.	Electronic signature requires validation. Forms based on Adobe forms.
Quality*Stream	Easy to install. Modern, rich GUI. Dashboards with quality metrics and corrective action repository.	Microsoft® SQL Server database isn't supported. Form loading

Software tool	Strengths	Weaknesses
	Monitors performance, integrity and reliability of business processes, across all enterprise applications. Robust business process management architecture and integration capabilities from Zaplet acquisition. Well suited for regulated multiplant environments with complex supply chains.	response time was sometimes too long.
Agile Advantage 2006	Product life-cycle management system for small and medium enterprises. Average Agile implementation completed in 12 weeks. Built-in DMS.	Microsoft® SQL Server database isn't supported. Expensive licensing model (e.g. 40 named user accounts for 10 concurrent users).

Table 2. Overview of the QMS software tools

In the product evaluation phase, three best ranked software vendors were selected to present their software solutions (CATSWeb®, TrackWise® and SmartSolve®) during the two day on-site workshops. Each vendor developed and set up its product prototype for the hands-on trial process (e.g. CAPA and document control process) and spent two days training users and answering lots of questions. Through the exercises, some hands-on experience and the discussions, participants (e.g. all relevant people including process owners, QA, IT and manufacturing personnel, etc.) developed their understanding of the different elements of a quality system (e.g. change control and document management, CAPA, complaints management, audit management, etc.). With hands-on access to the vendor tools, participants found that such sessions were valuable and helpful to evaluate the three vendor products. The selection team attended all demos and workshops and at the end of each workshop an evaluation questionnaire was used to obtain a formal feedback from all attendees.

Collected feedback from users in written format was assessed. Reference calls have been held with two medical device companies from Holland and Austria, for the products of Pilgrim Software and AssurX, respectively. Additionally one Company A employee reviewed on site the Sparta Systems TrackWise® implementation in one medical device company in Germany.

After the workshops had been completed it was clear that CATSWeb® product did not reach Company A requirements for quality management software. Therefore the short list was reduced to 2 vendors with a recommendation of the preferred solution. Based on time reduction estimates by the workshop participants, the ROI (return on investment) calculation was set up.

4.4. Product Selection Phase

The senior management reviewed the evaluation summary with other related evaluation documents and the most suitable quality management software, TrackWise®, was finally selected.

Problems which appeared during the quality management software assessment and selection process were:

- Lack of well defined evaluation process.
- The evaluation plan was not defined at the project start – The plan should include the resource and schedule estimates, stakeholder identification, evaluation approach, evaluation criteria, etc.
- Conducting vendor searches was a very time consuming process.
- The degree of user involvement in the requirements analysis and specification was not enough – Requirements were mainly defined by engineers with consulting process owners (i.e. end users).

5. Conclusion and Future Work

Successful selection of quality management software to fit required requirements is a critical and complex activity. It requires a significant investment of time and resources, top management support and commitment, the involvement of the entire organization, research, planning and re-evaluation. It should allow evaluating and selecting the software that best meets your requirements.

This paper presents a list of requirements structured in an evaluation framework for selecting quality management software tools. After appropriate modifications this evaluation framework might be suitable to assess other software tools as well, e.g. ERP tools. The results of the QMS tool evaluation show that there is no omnipotent tool for solving all problems.

The biggest challenge for project success is requirements gathering and documentation of requirements, more commonly called requirements analysis and specification. To avoid typical software selection mistakes, it is essential to use a structured, systematic selection and evaluation process and to define the requirements in the early selection phase.

Factors affecting purchasing decision are many and should include: pricing and total cost of ownership, return on investment, product functionality, hardware platform requirements, software compatibility (i.e. integration with other systems), service and support, technology and system architecture, vendor longevity, training, documentation, applicable standards (legal, regulatory, communications), multilingual capabilities, quality attributes (e.g. reliability, stability, security, performance, maintainability, testability, portability, availability, usability, reusability), legislative requirements (e.g. privacy, safety), etc. All factors should be considered before making a final business decision. Performing on-site demonstrations that end users can gain some hands-on experience is one of the most significant steps in any software evaluation and selection process.

References

[1] Alanbay, O. ERP Selection using Expert Choice Software. In *Proceedings of the 8th International Symposium on the Analytic Hierarchy Process (ISAHP)*, Honolulu, Hawaii, 2005.

[2] Alves, C.F. et al. Requirements Engineering for COTS Selection. *III Workshop on Requirements Engineering (WER 2000)*, pages 194-208, Rio de Janeiro, Brazil, 2000.

[3] Ayala Martinez, C.P. Systematic Construction of Goal-Oriented COTS Taxonomies. PhD thesis, Technical University of Catalunya, 2008.

[4] Bayazit, O.; Karpak, B. An AHP Application in Vendor Selection. In *Proceedings of the 8th International Symposium on the Analytic Hierarchy Process (ISAHP)*, Honolulu, Hawaii, 2005.

[5] Boehm, B.; Basili, V.R. Software Defect Reduction Top 10 List. *IEEE Software*, pages 135-137, Vol. 34, No. 1, 2001.

[6] Boehm, B.W.; Papaccio, P. N. Understanding and Controlling Software Costs. *IEEE Transactions on Software Engineering*, pages 1462-1477, Volume 14, Issue 10, 1988.

[7] Comella-Dorda, S.; Dean, J.; Morris, E.; Oberndorf, P. A Process for COTS Software Product Evaluation. In *Proceedings of the 1st International Conference on COTS-Based Software System*, pages 86-96, Orlando, Florida, Lecture Notes in Computer Science 2255, Springer-Verlag, 2002.

[8] Dean, J.C.; Vidger, M.R. COTS Software Evaluation Techniques, In *Proceedings of the NATO/RTO Information System Technology, RTO IST Symposium on*

Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS", paper published in RTP MP-48, Brussels, Belgium, 2000.

[9] FDA et al. *Guidance for Industry Part 11, Electronic Records; Electronic Signatures – Scope and Application*. Pharmaceutical cGMPs, USA, 2003.

[10] FDA. FDA 21 CFR Part 820 Quality System Regulation. Pages 138-151, 2006, http://www.gmp1st.com/82006.pdf, downloaded: May, 24th 2008.

[11] FDA. Glossary of Computerized System and Software Development Terminology. August, 1995, http://www.fda.gov/ora/inspect_ref/igs/gloss.html, downloaded: May, 24th 2008.

[12] FDA; CDRH. *Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices*. USA, 1999.

[13] IEEE. *IEEE Std 1233-1998 IEEE Guide for Developing System Requirements Specifications*. The Institute of Electrical and Electronics Engineers, Inc., IEEE Computer Society Press, New York, 1998.

[14] IEEE. *IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology*. The Institute of Electrical and Electronics Engineers, Inc., IEEE Computer Society Press, New York, 1990.

[15] IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. The Institute of Electrical and Electronics Engineers, Inc., IEEE Computer Society Press, New York, 1998.

[16] ISO. *ISO 13485:2003 Medical devices – Quality management systems – Requirements for regulatory purposes*. Geneva, 2003.

[17] ISO. *ISO/IEC 25030:2007 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements*. Geneva, 2007.

[18] ISO. *ISO/IEC 25051:2006 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing*. Geneva, 2006.

[19] ISO. *ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model*. Geneva, 2001.

[20] Kontio J. et al. A COTS Selection Method and Experiences of Its Use. In *Proceedings of the 20th Software Engineering Workshop*, NASA Software Engineering Laboratory, Greenbelt, Maryland, 1995.

[21] Kontio, J. A Case Study in Applying a Systematic Method for COTS Selection. In *Proceedings of the 18th International Conference on Software Engineering (ICSE)*, pages 201-209, Berlin, Germany, 1996.

[22] Kontio, J. *OTSO: A Systematic Process for Reusable Software Component Selection*. CS-TR-3478, Technical report, University of Maryland, College Park, Maryland, 1995.

[23] Kunda, D. *A social-technical approach to selecting software supporting COTS-Based Systems*. PhD thesis, University of York, Department of Computer Science, 2001.

[24] Kunda, D.; Brooks, L. Applying social-technical approach for COTS selection. In *Proceedings of the 4th UKAIS Conference*, pages 552-565, University of York, York, McGraw Hill, 1999.

[25] Liang, S.; Lien, C. Selecting the Optimal ERP Software by Combining the ISO 9126 Standard and Fuzzy AHP Approach. *Contemporary Management Research*, pages 23-44, Vol. 3, No. 1, 2007.

[26] Maiden, N.A.M.; Ncube, C. Acquiring COTS Software Selection Requirements. *IEEE Software*, pages 46-56, Vol. 15, No. 2, 1998.

[27] Miller, A. COTS Software Supplier Identification and Evaluation. In *Proceedings of the NATO/RTO Information System Technology, RTO IST Symposium on Commercial Off-The-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS"*, paper published in RTO MP-48, Brussels, Belgium, 2000.

[28] Morera, D. COTS Evaluation Using Desmet Methodology & Analytic Hierarchy Process (AHP). In *Proceedings of the 14th International Conference, PROFES 2002*, pages 485-493, Rovaniemi, Finland, Lecture Notes in Computer Science 2559, Springer-Verlag, 2002.

[29] Morisio, M.; Tsoukiàs, A. IusWare: A methodology for the evaluation and selection of software products. *IEEE Proceedings of Software Engineering*, pages 162-174, Vol. 144, No. 3, 1997.

[30] Ncube, C.; Dean, J.C. The Limitations of Current Decision-Making Techniques in the Procurement of COTS Software Components. In *Proceedings of the 1st International Conference on COTS-Based Software System*, pages 176-187, Orlando, Florida, 2002.

[31] Ncube, C.; Maiden, N.A.M. PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm. In *Proceedings of the 2nd International Workshop on Component-Based Software Engineering*, pages 1-12, Los Angeles, California, 1999.

[32] Saaty, T.L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw Hill, New York, 1980.

[33] Sheldon, F.T. et al. Reliability Measurement: From Theory to Practice. *IEEE Software*, pages 13-20, Vol. 9, No. 4, 1992.

[34] Smith, A. *Regulatory Compliance Reshapes the Quality Software Market*. AMR Research Report, AMR Research, Inc., 2004.

[35] The Standish Group. *Extreme CHAOS*. The Standish Group International Inc., February, 2001.

[36] The Standish Group. *The CHAOS Report*. The Standish Group International Inc., December, 1994.