A Knowledge-Based Advisory System for Software Quality Assurance

Khalid Eldrandaly
Information Systems Department, Zagazig University, Egypt

Abstract: *Software quality assurance is a planned and systematic approach to ensure that software processes and products confirms to the established standards, processes, and procedures. The goals of software quality assurance are to improve software quality by appropriately monitoring both software and the development process to ensure full compliance with the established standards and procedures. There are several models for software quality assurance, such as the ISO/IEC 90003, and the capability maturity model integration. However, the proper implementation of these models is often a difficult and a costly task for software companies especially small and medium ones. This paper describes a prototype knowledge-based advisory system designed to play the role of a "virtual quality editor" to help individuals, organizations, and software companies who desire to implement these quality models. By identifying the gauging absence of prerequisites between the prerequisites and what actually exists in the present environment, this advisory system provides assessment results and suggestions to the companies so that successful implementation of these quality models can be achieved. Component object model technology is used in designing and integrating the different components of the prototype system to assure software interoperability between these components. The architecture, the development and the implementation of the prototype system are discussed in details. A typical example is also presented to demonstrate the application of the prototype system.*

Keywords: *Expert systems, software quality assurance, ISO/IEC 90003, capability maturity model integration, component object mode.*

Received December 9, 2006; accepted April 28, 2007

1. Introduction

The quality of products and services has become one of the most important factors that influence national and international business [20]. Achieving a high level of product or service quality is now the objective of most organizations. It is no longer acceptable to deliver poor quality products and then repair problems and deficiencies after they have been delivered to the customer. In this respect, software is the same as any other manufactured product such as computers. The responsibility of quality managers is to ensure that the required level of quality is achieved. Quality management involves defining appropriate procedures and standards and checking that these are followed by all engineers [27]. Software Quality Assurance (SQA) is a planned and systematic approach to ensure that software processes and products confirms to the established standards, processes, and procedures. The goals of SQA are to improve software quality by appropriately monitoring both software and the development process to ensure full compliance with the established standards and procedures. Establishing standards and procedures for software development is critical, since these provide the framework from which the software evolves. Standards are the established criteria to which the software products are compared. Procedures are the established criteria to which the development and control processes are compared. Standards and procedures establish the prescribed methods for developing software; the SQA role is to ensure their existence and adequacy. Proper documentation of standards and procedures is necessary since the SQA activities of process monitoring, product evaluation and auditing rely upon unequivocal definitions to measure project compliance [22]. Quality management and assurance is a common concept in today's business. In fact, many companies lost their market share simply because of their flawed approach to quality management [20]. A vast majority of software producers, which haven not yet implemented a methodology for software quality assurance, are paying high costs of production and systems maintenance, and are therefore being displaced from the global market [15].

There are several models for software quality assurance, such as the ISO/IEC 90003, and the Capability Maturity Model Integration (CMMI). However, the proper implementation of these models is often a difficult and a costly task for software companies' especially small and medium ones. This paper describes a prototype knowledge-based advisory system designed to play the role of a "virtual quality editor" to help individuals, organizations, and software companies who desire to implement these quality models. By identifying the Gauging Absence of Prerequisites (GAP) between the prerequisites and what actually exists in the present environment, this advisory system provides assessment results and suggestions to the companies so that successful implementation of these quality models can be achieved. Component Object Model (COM) technology is used in designing and integrating the different components of the prototype system to assure software interoperability between these components. The architecture, the development and the implementation of the prototype system are discussed in details. A typical example is also presented to demonstrate the application of the prototype system.

2. ISO/IEC 90003:2004

International Organization for Standardization (ISO) is the world's largest developer of standards. It was set up in 1947 and is located in Geneva, Switzerland. The international standards which ISO develops are very useful because they contribute to making the development, manufacturing and supply of products and services more efficient, safer and cleaner. The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes international standards for all electrical, electronic and related technologies. It was set up in 1906 and is also located in geneva, switzerland. Both standards organizations are supported by national member bodies. These member bodies participate in the standards writing process through technical committees [18, 16].

ISO/IEC 90003:2004 was developed by ISO/IEC Joint Technical Committee1 (JTC1), Subcommittee7 (SC7). JTC1 is responsible for all kinds of information technology standards while SC7 is specifically responsible for the development of all software and system engineering standards. ISO/IEC 90003:2004 was officially published on february 15, 2004. This international standard provides guidance for organizations in the application of ISO 9001:2000 to the acquisition, supply, development, operation and maintenance of computer software and related support services. It does not add to or otherwise change the requirements of ISO 9001:2000. ISO/IEC 90003 2004 is not supposed to be used as a set of criteria against which software oriented quality management systems are to be assessed for certification (registration) purposes. It is not a certification standard. Instead, ISO 90003 is used to develop an ISO 9001 2000 quality management system which can be used to apply for an ISO 9001 certificate (not an ISO 90003 certificate). The following equation summarizes how ISO IEC 90003:2004 and ISO 9001:2000 are related [25]:

$$\textit{ISO 90003} = \textit{ISO 9001} + \textit{Advice on How to Apply ISO 9001}. \quad (1)$$

Detailed description of ISO/IEC 90003:2004 is reported elsewhere [24, 17].

3. CMMI

Capability Maturity Model® Integration (CMMI) was developed by the Software Engineering Institute of Carnegie-Mellon University under the sponsorship of the US department of defense. Beginning with the Capability Maturity Model for Software (SW-CMM) and now continuing with the Capability Maturity Model Integration (CMMI) framework, software development organizations have achieved significant gains in their ability to develop and deliver systems with predictable results [4]. A model is a simplified representation of the world. A maturity model is a structured collection of elements that describe characteristics of effective processes. A maturity model can be used as a benchmark for assessing different organizations for equivalent comparison. CMMI is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes.

There are multiple CMMI models available, as generated from the CMMI framework. Also, there are two types of representations in the CMMI models: staged and continuous. A representation allows an organization to pursue different improvement paths. Consequently, you need to be prepared to decide which CMMI model and which representation best fit your organization's process-improvement needs. In this research, we used CMMI for software engineering with staged representation. This model organizes process areas into five maturity levels to support and guide process improvement. The staged representation groups process areas by maturity level, indicating which process areas to implement to achieve each maturity level. Maturity levels represent a process-improvement path illustrating improvement evolution for the entire organization pursuing process improvement. The five maturity levels are initial, managed, defined, quantitatively managed, and optimizing. Detailed description of CMMI is reported elsewhere [5, 6].

4. SQA Advisory System

A prototype advisory system was developed using three COM-compliant commercially available software packages: Microsoft®Visual Basic 6.0, Visual Rule Studio®, Microsoft® Access 2003. Microsoft®Visual Basic 6.0 was used to provide the shell for the COM integration, and to develop the system's user interface. Visual Rule Studio® was used to develop the expert system module. Microsoft® Access 2003 was used to develop the database module.

The proposed system was developed as a three-tire architecture as shown in Figure 1. This software industry standard architecture provides a framework for logical components of the software to interact and enables flexibility in managing changes and updates in system components.

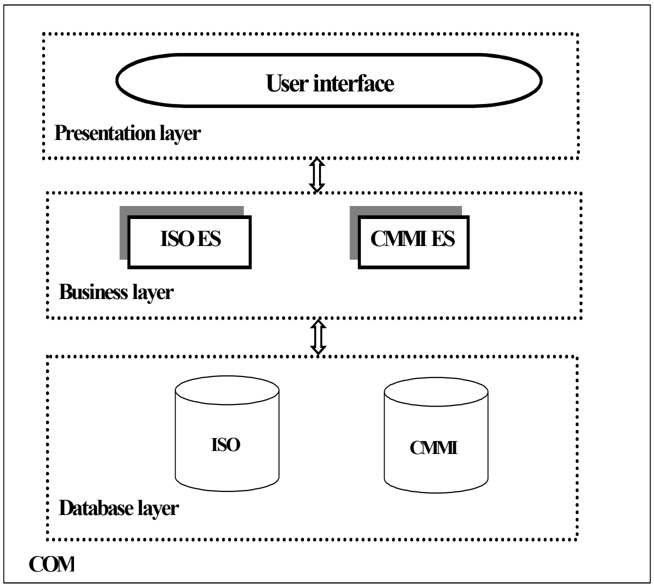

Figure 1. Three-tire architecture of the proposed system.

5. Development of the Prototype Expert System

Expert systems are fast becoming the leading edge of Artificial Intelligence (AI) technology because of the need for such systems in commercial and scientific enterprises and also because AI technology has evolved to the point where expert systems development has become well understood and feasible in many domains. An expert system is a computer program that embodies the expertise of one or more experts in some domain and applies this knowledge to make useful inferences for the user of the system [14, 28, 10, 2].

Expert system technology is beginning to play an important role and will become more common in the-future in quality management [1]. Reviewing the literature shows that several authors such as [8, 12, 29, 7, 9, 11, 23, 3, 19, 15, 20, 13] have provided interesting studies in this domain.

Visual Rule Studio® (an object-oriented COM-compliant expert system development environment for windows) was used to develop the prototype expert system. Visual rule studio solves the problem of software interoperability by allowing the developers to package rules into component reusable objects called rulesets. By fully utilizing OLE and COM technologies, rulesets act as COM automation servers, exposing ruleset objects in a natural COM fashion to any COM compatible client. Visual rule studio installs as an integral part of MS visual basic 6.0, professional or enterprise editions, and appears within the visual basic as an activeX designer. This allows the developers to add rule objects to their existing or new visual basic application in much the same manner they would extend their application with a new form or activeX control. rulesets can be complied within visual basic. EXE, .OCX, or .DLL executables and used in any of the ways the developers normally use such executables [26].

The visual rule studio's object-oriented rules technology is a new adaptation of rule-based expert system technology. It is based on the Production Rule Language (PRL) and inference engines of LEVEL5 object®. Rules in a production system consist of a collection of if condition-then action statements. The knowledge base of the proposed expert system consists of two different rulesets. The first ruleset consists of 1 class and 173 rules, and the second ruleset consists of 1 class and 163 rules.

The visual rule studio inference engine provides two primary problem-solving engines relevant to production systems: the forward chaining engine and the backward chaining engine. In the proposed expert system forward chaining engine is used. Starting from an initial or current set of data, the forward chaining inference engine makes a chain of inferences until a goal is reached. In forward chaining the data values of the context are matched against the IF parts, or left-hand-sides, of rules. If a rule's IF side matches the context, then the inference engine executes the Then part, or right-hand-side of the rule. If the execution of the Then part of a rule changes the data values of the context, then the inference engine repeats the entire match-execute cycle again using the new state of the context data values as a new initial set of data.

6. Database Module

Microsoft® Access 2003 was used to develop the database module. This module contains two different databases, one of them contains the questions and the assessment results of ISO and the other contains the questions and the assessment results for CMMI.

Microsoft® ActiveX® Data Object (ADO) was used to read the required information from the database and write the assessment results in the database. ADO was implemented using a set of COM-based interfaces that provide applications with uniform access to data stored in diverse information sources [21].

7. Example of Consultation Session

In order to demonstrate how the proposed system can be used as a virtual auditor for self assessment of software companies, a sample run is demonstrated in this section.

Upon execution of the system, the main screen appears as shown in Figure 2. By pressing the start button, a new screen appears that gives the user the opportunity to select the required software quality model as shown in Figure 3. Upon choosing the software quality model, the system asks the user to enter or to create his username and password as shown in Figure 4. After that the system gives the user the opportunity either to create a new project or to choose one from the database as shown in Figure 5. Then the user has the opportunity either to evaluate the whole quality levels or to evaluate a level by level as shown in Figure 6. If the user chooses to evaluate a level by level, then the system gives the user the opportunity to choose the required level as shown in Figure 7. After choosing the required level the system begins to ask the user the related assessment questions as shown in Figure 8. Upon finishing all the related questions, the user has the opportunity to see the assessment report as a text, a bar graph or a pie graph as shown in Figures 9, 10, 11, and 12.

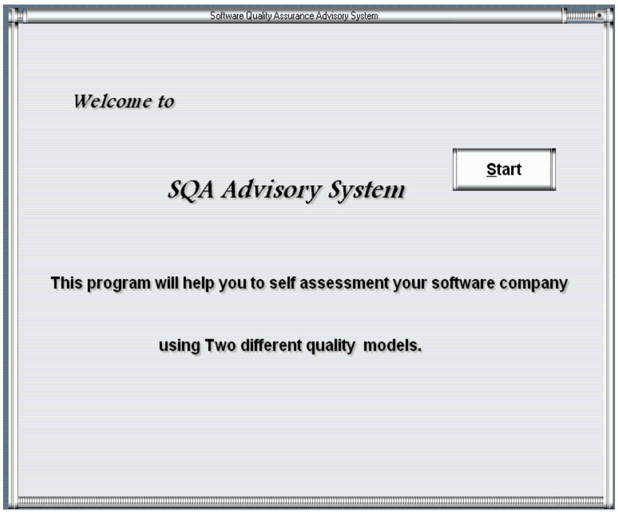

Figure 2. Main screen.

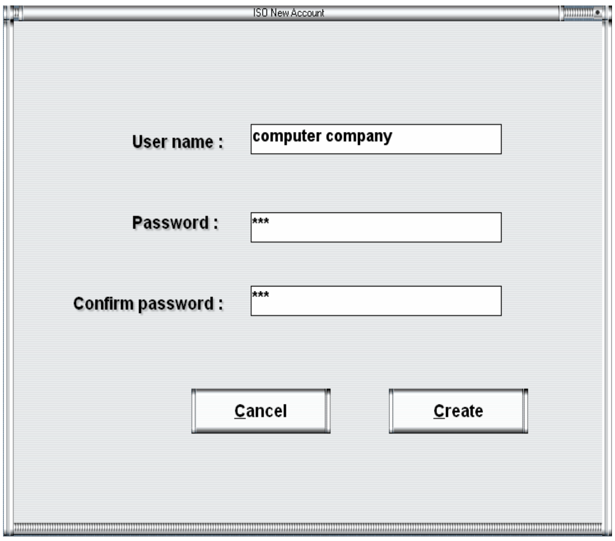

Figure 3. Choosing the quality model.

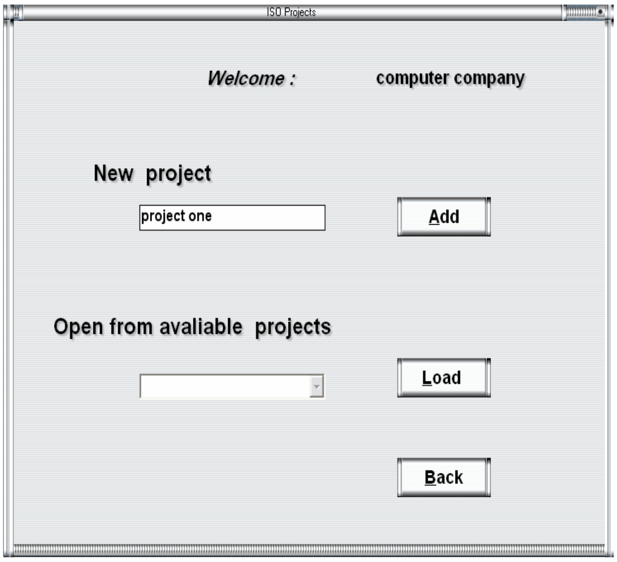

Figure 4. Entering or creating the username and password.

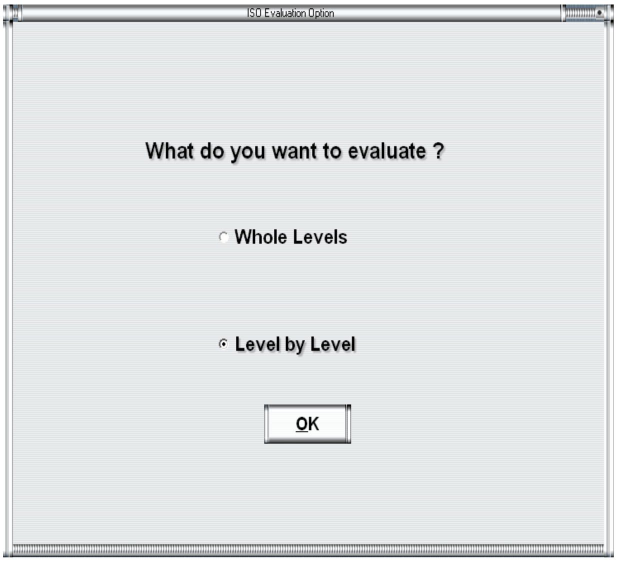

Figure 5. Loading or creating a project.

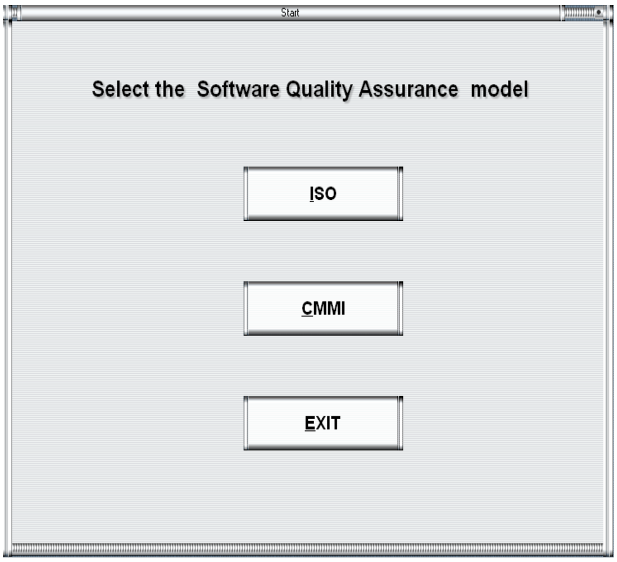

Figure 6. Choosing the evaluation style.

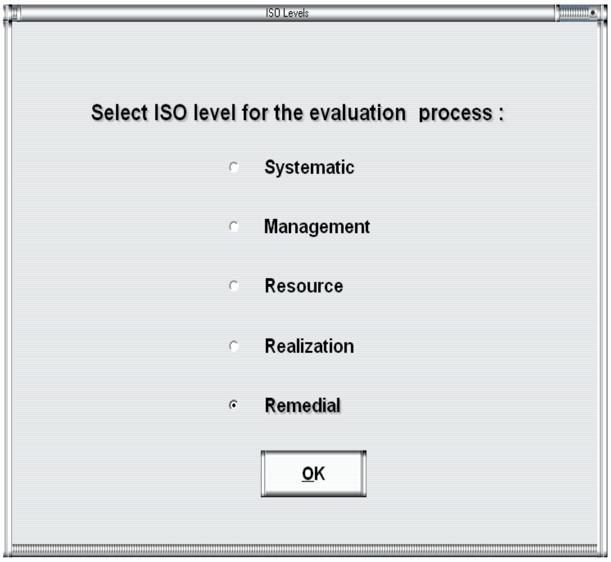

Figure 7. Choosing the quality level.

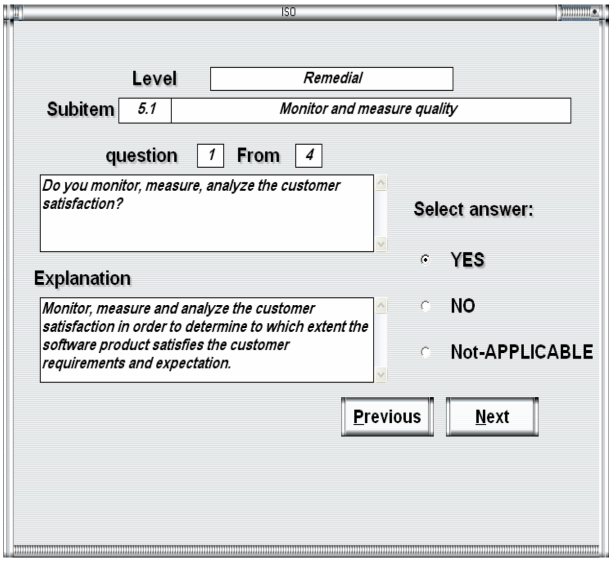

Figure 8. A sample of ISO questions.

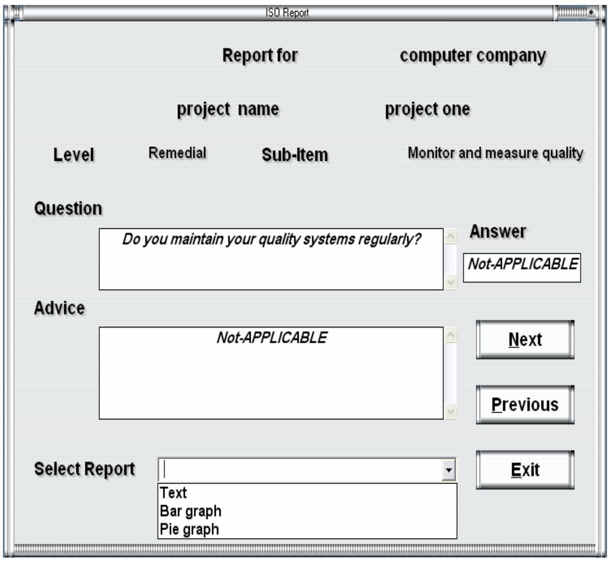

Figure 9. Choosing the assessment report style.

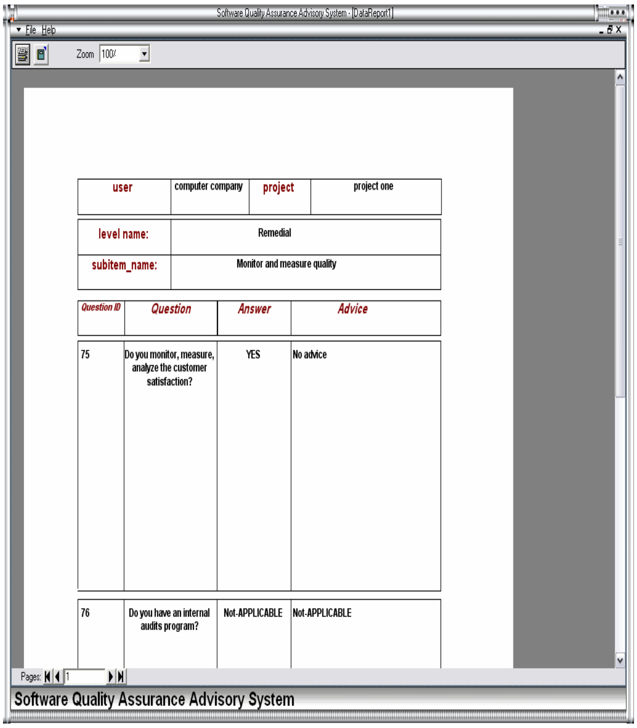

Figure 10 A text report.

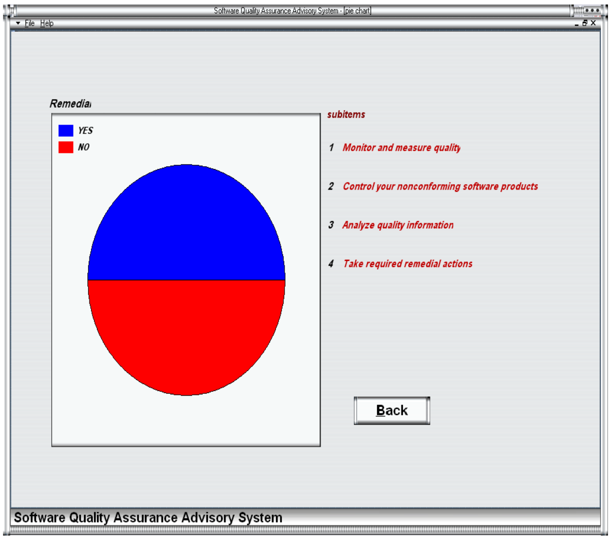

Figure 11. A pie chart report.

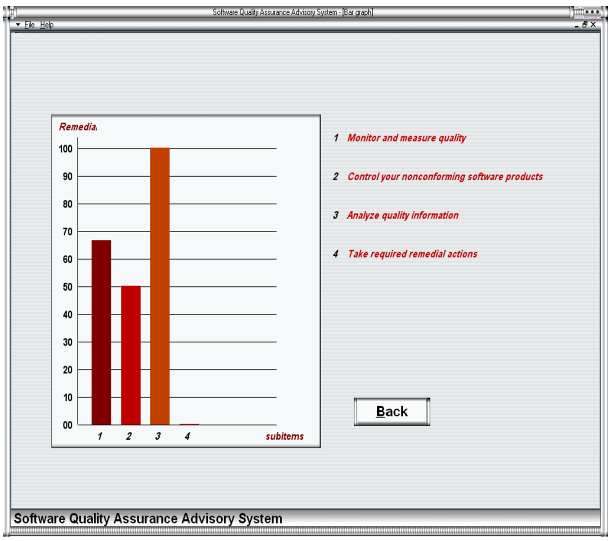

Figure 12. A bar chart report.

8. Conclusion

In this paper, a prototype knowledge-based advisory system for software quality assurance is presented. This system is designed to play the role of a "virtual quality editor" to help software companies to implement ISO and CMMI quality models. By identifying the Gauging Absence of Prerequisites (GAP) between the prerequisites and what actually exists in the present environment, this advisory system provides assessment results and suggestions to the companies so that successful implementation of these quality models can be achieved. Also, this system could be used for training novice auditing personnel. The architecture, the development, and the implementation of the prototype system are discussed in details. The use of visual rule studio® (an object-oriented COM-compliant expert system development environment for windows) which runs together with microsoft visual basic 6.0 is found to be very effective in producing the system under Windows environment. Also, software interoperability between the different components of the system is achieved by adopting the COM technology in designing the system. The system can be easily modified to include another software quality assurance models such as the Software Process Improvement and Capability dEtermaination (SPICE).

References

[1] Allen B. and Kathawala Y., "Expert systems Applications in Quality Management: An Integrative Approach," *ISA Tran*sactions, vol. 31, no. 2, pp. 19-25, 1992.

[2] Awad E., *Building Knowledge Automation Expert Systems with EXSYS CORVID*, EXSYS Inc, Albuquerque, USA, 2003.

[3] Bayraktar D., "A Knowledge-Based Expert System Approach for the Auditing Process of Some Elements in the Quality Assurance System," *International Journal of Production Economics*, vol. 56-57, pp. 37-46, 1998.

[4] Carnegie Mellon University Software Engineering Institute, *CMMI Overview*. Pittsburgh, USA, 2005.

[5] Carnegie Mellon University Software Engineering Institute, *CMMI-SW Vesrsion1.1: Staged Representation.* Pittsburgh, USA, 2002.

[6] Chrissis M., Konrad M., and Shrum S., *CMMI*: *Guidelines for Process Integration and Product Improvement*, Addison-Wesley, 2003.

[7] Chung K. and Tou J. "Knowledge-Based Approach to Fault Diagnosis and Control in Distributed Process Environments," *in Proceedings of the International Society for Optical Engineering (SPIE)*, USA, pp. 323-331, 1991.

[8] Crawford K. and Eyada O., "A Prolog Based Expert System for the Allocation of Quality Assurance Program Resources," *Computers and Industrial Engineering*, vol. 17, no. 14, pp. 298-302, 1989.

[9] Crossfield R. and Dale B., "The Use of Expert Systems in Total Quality Management: An Exploratory Study," *Quality and Reliability Engineering International*, vol. 7, pp. 19-26, 1991.

[10] Darlington K., *The Essence of Expert Systems*, Prentice-Hall, 2000.

[11] Edgell N. and Kochhar A., "BS 5750 Quality Assurance System: A Knowledge Based Approach to System Audit and Implementation," *IEE Conference Publication*, no. 359, Michael Faraday House, Stevenage, England, 1992.

[12] Eyada O., "An Expert System for Quality Assurance Auditing," *1990 ASQC Quality Congress Transactions*, vol. 44, pp. 613-619, 1990.

[13] Goldebrg Y. and Shmilovici A., "An Expert System Approach for Quality Assurance Auditing," *International Journal of Advanced Manufacturing Technology*, vol. 26, pp. 415-419, 2005.

[14] Hayes-Roth F., Waterman D., and Lenat D., *Building Expert Systems*, Addison-Wesley, 1983.

[15] Herrera E. and Ramirez R., "A Methodology for Self-Diagnosis for Software Quality Assurance in Small and Medium-Sized Industries in Latin America," *Electronic Journal of Information Systems in Developing Countries (EJISDC),* vol. 15, no. 4, pp. 1-13, 2003.

[16] IEC, *About the IEC*, http://www.iec.ch/about/mission-e.htm, 2006.

[17] ISO, *ISO/IEC 90003:2004*: *Software Engineering Guidelines for the Application of ISO 9001:2000 to Computer Software*, ISO Press, 2004.

[18] ISO, *Overview of the ISO*, http://www.iso.org /iso/en/aboutiso/introduction/index.html, 2006.

[19] Khan M. and Hafiz N., "Development of an Expert System for Implementation of ISO 9000 Quality Systems," Total Quality Management, vol. 10, no. 1, pp. 47-59, 1999.

[20] Liao H., Enke D., and Wiebe H., "An Expert Advisory System for ISO 9001 Quality System," *Expert Systems with Applications*, vol. 27, pp. 313-322, 2004.

[21] Microsoft, *Microsoft Developer Network Online Documentation*, MSDN Library, 2003.

[22] NASA Software Quality Assurance Center, *Software Assurance Guidebook-NASA-GB-A201*, 1989.

[23] Oppermann R. and Reiterer H., "Software Evaluation Using the 9241 Evaluator," *Behaviour & Information Technology*, vol. 16, no. 4/5, pp. 232-245, 1997.

[24] Praxiom Research Group Limited, Introduction to ISO IEC 90003 2004, http://www.praxiom.com/iso-90003-intro.htm, 2006.
[25] Praxiom Research Group Limited, *ISO IEC 90003 2004 Software Standards in Plain English*, Canada, 2004.
[26] RuleMachines, *Visual Rule Studio Developer's Guide*, Canada: OnDemand, 2002.
[27] Sommerville I., *Software Engineering*, Addison-Wesley, 2001.
[28] Waterman D., *A Guide to Expert Systems*, Addison-Wesley, 1986.
[29] Willborn W., "Expert Systems in Support of Quality Management," *1990 ASQC Quality Congress Transactions*, vol. 44, pp. 758-763, 1990.

Khalid Eldrandaly received his MS in systems engineering (expert systems) and his PhD in systems engineering (GIS). He was a visiting scholar at Texas A & M University, USA, for two years. Currently, he is an assistant professor of computer information systems and interim head of Information Systems and Technology Department, College of Computers and Informatics, Zagazig University, Egypt. His area of interests includes GIS, expert systems, SDSS, MCDM, and intelligent techniques in decision making. He is a member of the World Academy of Young Scientists, Arab Union of Scientists Researchers, Texas A&M International Faculty Network, and Egyptian Software Engineers Association.