

## **KNOWLEDGE MANAGEMENT IN SOFTWARE DEVELOPING ORGANIZATIONS. COMBINING CONCEPTUAL MODELS**

*Karol Chrabański, The General Jerzy Ziętek Silesian School of Management in Katowice, chrabanski@omi.pl*

### **ABSTRACT**

*Development of software provided by Software Developing Organizations (SDO) necessitates the improvement of Quality Management System (QMS), which constitutes the basis for software design. Software is a product which is almost entirely made up of knowledge. It seems that combining Quality Management System (QMS) and Knowledge Management System (KMS) in Software Developing Organizations may contribute to providing final users with the software which meets their needs to a higher degree and does not contain errors. The article presents the prerequisites for integration and the proposed way of combining QMS and KMS systems. The underlying concept of the paper is the author's assumption concerning similarities between the structures of both systems. The conceptual models, which are created and analyzed in the study, are later used to verify this hypothesis.*

**Keywords:** Software Developing Organization, Knowledge Management, Quality Management Systems, Conceptual models

### **INTRODUCTION**

The importance of software for proper functioning and development of contemporary civilization is indisputable. The findings of numerous authors lend support to the claim. For example, K. Khatatneh and T. Mustafa (2009) argue that: "Software has become an essential part of many industrial, military and even commercial systems" (p.154). In addition, researchers have long pointed to the fact that in order to meet the clients' requirements, software is becoming more and more complex. [Le Du & Challet, 2005.]. The consensus view seems to be that failures of complex systems are caused by failures of software. However, the errors which are detected and eliminated in time may contribute to developing programming practices and assuring better software quality, among others, by preventing the appearance of such effects [Wallace & Kuhn, R. 2001], [Suma & Gopalakrishnan, 2008]. It seemed that the solution to this problem would be to undertake attempts to devise different models which would focus on finding a specific "pathway" to ensure the creation of safe and reliable software. However, this should always be carried out in line with software engineering and with the consideration of all – presently known – factors which might affect its quality [Aisultanny & Wohaiishi, 2009]. The models which aim at implementing the above notion of creating a "pathway" to ensure safe and reliable software are presented below.

#### **Model 1:**

Focus on the analysis of the feedback content, where errors and control reports were one of the sources of information [Amadio, 2005], [Perry, 2005]

#### **Model 2:**

Display of the overview of software and regarding it as a fundamental tool ensuring quality [Olalekan & Adenike, 2008]

#### **Model 3:**

Proposal to make testing the key to eliminating software errors [Park, Hwang & Fujiwara, 2005]

#### **Model 4:**

Pointing to the need of integrating TQM (Total Quality Management) with IT auditing system [Shang & Lin, 2009], [Sureshchandar & Leisten, 2006], [Chou & Chen, 1998]

**Model 5:**

Assuming that improving the quality of software may be performed by determining an expected number of errors in particular modules. The efforts undertaken by software designers–programmers are focused on the modules where the greatest risk of failure appears. Elimination or reduction of the expected number of errors in those modules should contribute to the improvement of software quality [Khoshgoftaar, Gao & Szabo, 2005].

In the summary review of some models, the findings of the study point to the fact that in different cases emphasis may be placed on various elements of the lifecycle of the programming product.

Even an initial review of the literature on software engineering allows – following the opinions of some authors – formulating the observation that better processes in the realization of software lifecycle lead to higher quality of the end product and its lower development costs [Arora, Caulkins & Telang, 2006]. Most frequently software producers have applied the available ISO 9000 standard. However, most of them were aware that, as A. Y. Aisultanny and A. M. Wohaishi (2009) claim: “ISO 9001 is an international quality management system standard applications to organizations in all types of businesses” (p.578). Therefore, the requirements of ISO 9001:2000 standards have been supplemented with the guidelines concerning its computer software applications [13]. The guidelines were designated as ISO/IEC 90003:2004 [14].

In the light of the author’s practical experience (designing information systems, devising quality management systems, testing the compliance of the implemented and maintained QMS in SDOs with the requirements of ISO 9001:2000 and ISO 9001:2008, etc.) and relevant publications on the subject, the following reasoning seems to have a sound basis. Since software is a product of knowledge, the processes of developing software should be linked to knowledge. The designed, documented, implemented and maintained KMS in SDO contributes to the development of software of proper quality (which is understood as meeting the clients’ needs). Developing QMS in SDO and establishing the principles of their application and development require relevant knowledge. In turn, the relevant knowledge provides for producing software of higher quality.

**QUALITY MANAGEMENT SYSTEMS AND KNOWLEDGE MANAGEMENT SYSTEMS IN SOFTWARE DEVELOPING ORGANIZATIONS. DEFINITIONS**

The models presented above focused on different elements which were to contribute markedly to ensuring the proper quality of the developed software. They did not indicate the direct links to knowledge. This, however, does not mean that in the relevant literature there are no independently defined concepts of quality [Merilinnä & Niemelä, 2005], [Khoshgoftaar, Bullard & Gao, 2009] and knowledge [Bento, Bento & White, 2014]. The discrepancies between the approaches to defining knowledge and quality are worth considering. In the case of quality we are dealing with uniform definitions, due to, for instance, the development and popularity of appropriate standards such as ISO 9000 [12], [West, 2007]. Following the indicated standards we may assume that “quality management system is a management system (set of interrelated or interacting elements to establish policy and objectives and to achieve those objectives) used to direct and control an organization in terms of quality” [12]. However, in the case of defining knowledge, such an approach was not adopted. Along these lines, R. Chalmers and R. Grangel (2008) argue that: “There is no universally accepted definition of exactly what knowledge is” (p.743). Tsai, Tsai & Lin (2012) add: “Management literature so far still lacks an acknowledged definition for knowledge” (p.24).

In the literature on the subject and in economic practice there are no doubts as to the importance of knowledge at the current stage of the civilization development, especially with regard to the emergence of today’s knowledge-based society. Here, we should quote the opinion expressed by J.A. Pratt and K. Hauser (2012): “Associated with the global trend towards knowledge-based societies, knowledge management has become increasingly important for companies to gain a competitive advantage in today’s marketplace” (p.9)

In this article, the author does not engage in polemics concerning the definitions of such concepts as knowledge, knowledge management or knowledge management system. The following definitions are adopted:

- i- Dingsøy and Conradi (2002): “Knowledge is then often defined as information that is used (in artificial intelligence – sense: in a computer system)” (p.305) Thus, the definition of knowledge combines information with its practical application
- ii- knowledge management is a practice which finds valuable information and transforms it into necessary knowledge, which is critical to decision-making processes and taking action [Van Beveren, 2002]. Due

to the fact that the work undertaken with regard to developing software takes the form of projects, this aspect is sometimes reflected in its definition. According to Carrillo and Chinowsky (2006), its wording might be as follows: “Knowledge management (KM) seeks to formalize the manner in which companies exploit their knowledge assets by harnessing organizational knowledge, promoting greater collaboration between groups with similar interests, capturing and using lessons learned on previous projects, etc.” (p.2).

- iii- knowledge management system represents processes aimed at finding valuable information. They can include localizing, acquiring, developing, sharing, using, and maintaining [Shang & Lin, 2009]

The author of the article has formulated and presented the methodology of combining the two systems. The presented methodology may be briefly described in the following way. The methodology is oriented to undertaking activities which transform the QMS (quality management system) into a frame of the KMS (Knowledge Management System) [Chrabański, 2013]. Suppositions that such solutions are feasible have already appeared in the literature on the subject [Carrillo & Chinowsky, 2006].

This article presents an outline of the approach focusing on creating Knowledge Management Systems (KMSs) in Software Development Organizations (SDOs). It is based on the solutions adopted in QMS, which are compliant with series 9000 ISO standards as well as its complementary form of ISO/IEC 90003:2004.

### **CONCEPTUAL MODELS. RESEARCH METHODOLOGY**

In the relevant literature, there emerge interesting concepts, which are, to a certain extent, consistent with the research carried out by the author. They may include the proposal presented by M. Linvall, I. Rus, and S.S. Sinha in the form of “KM Architecture Model”, which emphasizes that: “The organization of knowledge is based on a corporate taxonomy and serves as a ‘knowledge map supported by classifying and indexing tools’ ” (p.20).

The approach presented by the author introduced several classifications and enabled, among others, the introduction of knowledge maps to be included in the publications on the subject.

This approach consists of several stages, and each of them presents specific classifications and maps. The stages of the presented approach are listed below [Chrabański, 2013]:

**Stage 1:** Classifying QMS processes

**Stage 2:** Developing assumptions of a semantic model for presentation of requirements of ISO 9001:2000 and recommendations of ISO/IEC 90003:2004 (hereinafter called “model”)

**Stage 3:** Presenting individual QMS processes by means of a semantic model.

**Stage 4:** Working out maps of individual processes while taking into account the model assumptions.

**Stage 5:** Working out potential decisions as a method for knowledge localization.

**Stage 6:** Defining knowledge elements of the KMS.

**Stage 7:** Analyzing knowledge elements of the KMS.

It appears that the stage of potential decisions, which would lead to the localisation of knowledge, requires an additional explanation. If we assume that knowledge is linked to a human, then – as J. Kisielnicki writes – decision-making in the process of management (here: knowledge management) is the basis of all management activity (planning, organizing, motivating, control) [Kisielnicki, 2008]. A similar approach is presented by K.M. Wiig (2003). As Wiig claims: “From a KM perspective, understanding of personal and organizational situation-handling, including decision-making and problem-solving, is important to manage knowledge successfully” (p.4).

Particular attention should be paid to stage 5. It needs to be emphasized that in the literature on the subject occasionally there appear different notions which represent rather similar concepts. One of them is a concept of

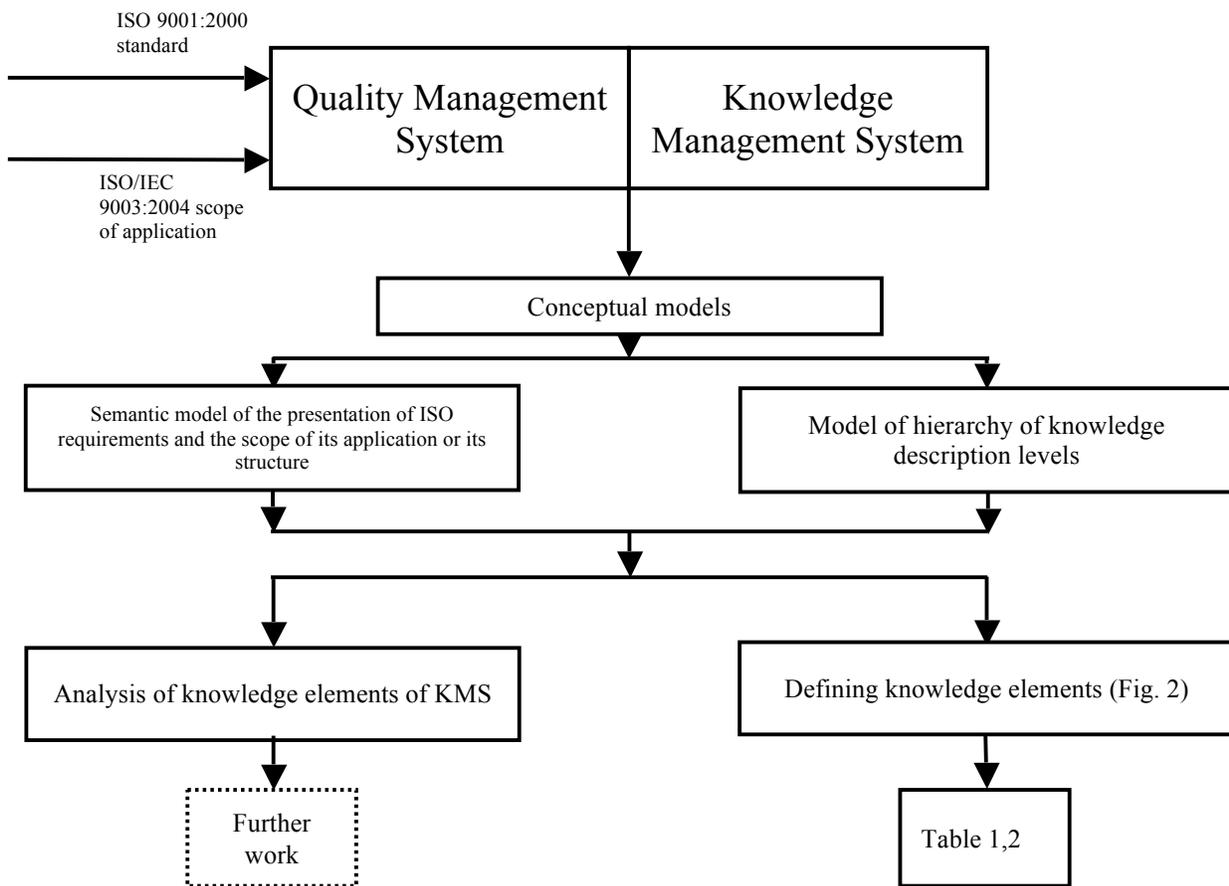
“architectural knowledge” which, according to Farenhorst, Lago and Van Vliet, “(...) is defined as the set of design decisions, including the rationale for these decisions, together with the resulting architectural design” (p.413). Stages 1 to 5 were described in the article included in References [Chrabański, 2013]. The present paper focuses on stage 6 (Defining knowledge elements of the KMS)

The research hypotheses which are to be tested in this study are as follows:

H1: The structures of QMS models and KMS models are similar (the number of levels)

H2: The combination of structures of QMS model and KMS model allows for the localization knowledge in SDO

The general concept is presented in Figure 1.



**Figure 1.** Conceptual Models of KMS and QMS

The starting point of the present considerations is the relation between QMS and KMS. Each of them is ascribed conceptual models, which enable looking into the structure of both systems. In the literature there are many examples of similar models, e.g.: a model proposed by Y. Sun and A. Bhattacharjee called: “Core IT usage model” [Sun & Bhattacharjee, 2014] or the model called Business Impact of Strategic Performance Measurement Systems, presented by A. Bento and his co-workers [Bento, Bento & White, 2014].

For quality management system a conceptual model may be both a semantic model and the ISO 9000 structure. The semantic model has been presented in the author's 2013 publication in the following way [Chrabanski, 2013]: "The semantic model should facilitate application of both the above standards through defining the meaning of contents of specific elements of the standards. They have been divided into the following groups:

- a) postulates, i.e. demands or requirements, and
- b) questions, i.e. issues that should be additionally considered and, if possible, resolved.

Both the postulates and the questions can be attributed with different meanings of the attached contents (extensions) as provided by the ISO 9001:2000 standard and the ISO/IEC 90003:2004 recommendations. Those extensions may refer to:

- a) the proposed method of realization,
- b) specification of the scope,
- c) additional notes as regards the realization method,
- d) examples,
- e) references to other sections of ISO 9001:2000,
- f) references to other standards." (p.397 )

In the present article – due to its limited volume – the semantic model will not be further analyzed. The author will apply conceptual model conformant with the structure of ISO 9001:2000 standard, including the recommendations of ISO/IEC 90003:2004. Its fragment is given in Tables 1 and 2 provided in the Annex.

The structure of KMS is presented with the model of hierarchy of knowledge description levels devised by K.M. Wiig, popularized in Polish literature by J. Gołuchowski [Gołuchowski, 2005]. It consists of the following elements: knowledge domain, knowledge region, knowledge segment, knowledge element, knowledge fragment and knowledge atom.

The author presented two conceptual models above. One model represents QMS and the other depicts KMS. These models will be combined. The elements of the hierarchy of knowledge description levels will constitute the combination levels of these models. These are: knowledge domain (level-1), knowledge region (level-2), knowledge element (level-3), knowledge fragment (level-4) and knowledge atom (level-5). Each of these levels corresponds to the elements of quality management systems. These items make up its structure. The author applied the structure of ISO 9001:2000 and ISO/IEC 90003:2004. Figure 2 demonstrates the numerical indications applied in ISO 9001:2000 and ISO/IEC 90003:2004. Due to the fact that ISO/IEC 90003:2004 standards are connected with ISO 9001:2000, in further analyzes the standards ISO 9001:2008 were not considered. This exclusion does not affect the conducted analysis in any significant way due to small differences between the two series with regard to their structure.

In order to document the compatibility of both models, the 7.3 standard (design and development) was selected. This choice was made for the following reasons:

- I. Design and development are probably the most difficult stages to implement in SDO. Hence, its structure has been described in detail.
- II. Design and development is an element of the lifecycle of an information system, which is implemented by each team designing and improving information systems.
- III. At the stage of design and development various information obtained by the team of software designers–programmers is used in practice. Hence, the assumption arises that information may be transformed into knowledge.
- IV. During the process of designing and development of information systems members of software designers–programmers' team make decisions, and the decisions localize knowledge which may be analyzed.
- V. Localized knowledge should be analyzed, in particular, for the sake of improving the effectiveness of the work conducted in the field of design and development, due to the fact that the errors made at this stage frequently result in unreasonable costs in the subsequent steps of the process.

An example of linking QMS with the hierarchy of knowledge description levels is presented in Figure 2. Key for Figure 2 is provided in the Annex (Table 1 and 2).

## CONCLUSION

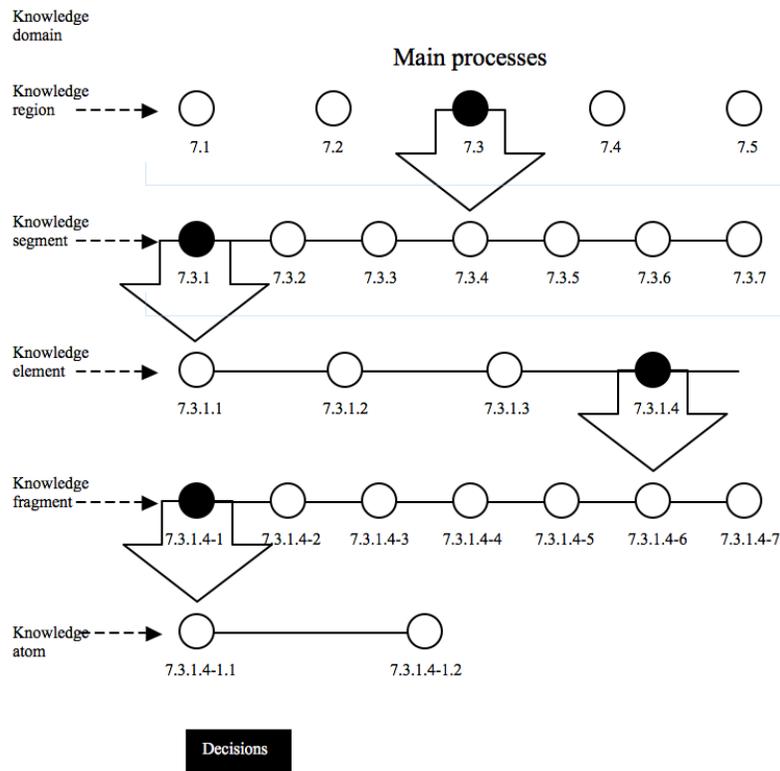
It appears that further development of the process consisting in generating and improving software, carried out by SDO, requires a combination of two systems, namely, quality management systems and knowledge management system. The said connection seems possible due to similar structures of both systems. The proposed hypothesis that the structures of QMS model and KMS model for SDO are similar is true (H1). The designed conceptual models, which reflect the structures of both systems, are the tools which help to verify this hypothesis.

Thus, in the case of the quality management system, the model structure may be the structure of 9001:2000 and ISO/IEC 90003:2004. The semantic model of the presentation of the requirements of ISO and the scope of its application may constitute the relevant structure. The author decided not to include it, considering the need to reduce the volume of the article. However, in the case of knowledge management system the author used the model of the hierarchy of knowledge description levels which is already available in the literature on the subject.

The defined structures are harmonized (they represent the same number of levels). The levels of the knowledge description have their counterparts in the elements of the structure of quality management systems. What is more, the levels of the knowledge description are presented in detail, based on the extension of the points of ISO 9001:2000 and ISO/ IEC 90003:2004 standards.

Attention should be paid to the fact the result of the above operation is the specific convergence which may be observed in the study, namely, the hierarchy of knowledge description levels indicates potential decisions localizing knowledge (H2). The content of these potential decisions shows the context of the abovementioned standards. The decisions undertaken within the framework of a quality management system constitute the atoms of knowledge in the hierarchy of knowledge description levels. Hence, the combination of the structures of QMS model and KMS model makes it possible to localize knowledge in software developing organizations. Thus, the hypothesis H2 has been positively verified.

Knowledge atoms, which are the decisions made, for example by information system designer, may be subject to various analyses.



### LIMITATIONS

Undoubtedly, the presented conceptual models are used to codify knowledge allocated to particular participants of the process of designing and developing information systems (software designers–programmers, project managers). We need to consider the limitations connected with the possibility of codifying knowledge. As T. Dingsoyr and R. Conradi (2002) claim: “We should add here that the codification strategy does not fit all types of knowledge. In situations where knowledge is very context-dependent, and where the context is difficult to transfer, it can be directly dangerous to reuse knowledge without analyzing it critically” (p.396).

Conceptual models in the scope of management information systems apply the requirements of ISO 9001:2000 and ISO/IEC 90003:2004 recommendations. ISO 9001:2008, as far as its structure is concerned – in its basic elements – does not differ from ISO 9001:2000. However, ISO 9001:2015, which is to be implemented, will probably have a different structure.

### APPLICATION POSSIBILITIES

The convergence of models of quality management systems and knowledge management systems may be used in further research, which would provide for developing quality management systems and knowledge management systems in software developing organizations. Potential decisions identified in the process of design and development of information systems help to reconstruct the decision-making process. Reconstruction will first be applied to the decisions made by an individual (software designer–programmer). "Patterns of decision-making" which can be recommended for use in the organization are likely to emerge. Also, another advantage of the presented approach is the ability to create and analyze the profile of the project manager, taking into account the criterion of effectiveness of project management (achieved objectives, resulting quality, and costs). In the long term, it will also be possible to develop knowledge maps for SDOs.

### REFERENCES

- Aisultanny, Y., A., & Wohaiishi, A., M. (2009), Essential Characteristics of Software Model that Provide the Software Quality Assurance. *International Review on Computers and Software*, 4(5), 577-578.
- Amadio, T. (2005). When Tool Licensing Affects Software Quality: Mitigating The Risk. *The Journal of the Quality Assurance Institute*, January, 6-10.
- Arora, A., Caulkins, J., & Telang, R. (2006), Research Note-Sell First, Fix Later: Impact of Patching on Software Quality, *Management Science*, 52(3), 465-471.
- Bento, A., Bento, R., & White, L., F. (2014), Strategic Performance Management Systems: Impact on Business Results, *Journal of Computer Information Systems*, Spring, 25-33.
- Carrillo, P., & Chinowsky, P. (2006), Exploiting Knowledge Management: The Engineering and Construction Perspective, *Journal of Management in Engineering* © ASCE, 2-10.
- Chalmeta, R., & Grangel, R. (2008), Methodology for the Implementation of Knowledge Management Systems, *Journal of the American Society for Information Science and Technology*, 59(5), 742-755.
- Chou, D., Yen, D., & Chen, J. (1998), Analysis of the Total Quality Management-Based Software Auditing *Total Quality Management*, 9(7), 611-618.
- Chrabański, K., (2013), Knowledge Management in Software Developing Organizations ; A New Paradigm *Issues in Information Systems*, 14(1), 394-402.

- Dingsøyr, T., & Conradi, R. (2002), A Survey of Case Studies of the Use of Knowledge Management in Software Engineering, *International Journal of Software Engineering and Knowledge Engineering*, 12(4), 391-414.
- Farenhorst, R., Lago, P., & Van Vliet, H. (2007), Eagle: Effective Tool Support for Sharing Architectural Knowledge, *International Journal of Cooperative Information Systems*, 16(3& 4), 413-437.
- Gołuchowski, J. (2005), Technologie informatyczne w zarządzaniu wiedzą w organizacji, [*Information technology in the management of knowledge in organizations*]. Katowice, Akademia Ekonomiczna Katowice, 31.
- ISO 9000:2000, Quality management systems – Fundamentals and vocabulary
- ISO 9001:2000, Quality management systems – Requirements
- ISO/IEC 90003:2004, Software engineering - Guidelines for the application ISO 9001:2000 to computer software
- Khatatneh, K., & Mustafa, T. (2009), Software Reliability Modeling Using Soft Computing Technique. *European Journal of Scientific Research*, 26(1), 154-160.
- Khoshgoftaar, T., & Bullard, L., A., Gao, K. (2009), Attribute Selection Using Rough Sets in Software Quality Classification, *International Journal of Reliability, Quality and Safety Engineering*, 16(1), 73–89.
- Khoshgoftaar, T., Gao, K., & Szabo, R. (2005), Comparing Software Fault Predictions of Pure and Zero-Inflated Poisson Regression Models, *International Journal of Systems Science*, September, 36(11), 705-715.
- Kisielnicki, J. (2008), "Jak zarządzać i być zarządzanym" [*Knowledge Acquisition and Management System*] in: Zarządzanie wiedzą we współczesnych organizacjach [*Knowledge Management in Contemporary Organizations*]. Warsaw, PWE, 61.
- Le Du, Y., & Challet, D. (2005), Microscopic Model of Software Bug Dynamics: Closed Source Versus Open Source *International Journal of Reliability, Quality and Safety Engineering*, 12(6), 522.
- Lindvall, M., Rus, I., & Sinha, S., S. Technology Support for Knowledge Management, [https://link.springer.com/chapter/10.1007%2F978-3-540-40052-3\\_9#page-1\\_\(17-03-2016\)](https://link.springer.com/chapter/10.1007%2F978-3-540-40052-3_9#page-1_(17-03-2016)) ).
- Merilinna, J., & Niemelä, E. (2005), A Stylebase as a Tool for Modelling of Quality-Driven Software Architecture, *Proceedings of the Estonian Academy Sciences Engineering*, 11(4), 296–312.
- Mundra, N., Karishma, G., & Vashisth, R. (2011), "Achieving competitive advantage through knowledge management and innovation: Empirical evidences from the Indian IT sector," *The IUP Journal of Knowledge Management*, 9(2), 7-25.
- Olalekan, A., & Adenike, O. (2008), Empirical Study of Factors Affecting the Effectiveness of Software Inspection: A Preliminary Report. *European Journal of Scientific Research*, 19(4), 614-627.
- Park, J.-Y., Hwang, Y.-S., & Fujiwara, T. (2005), Integration of Imperfect debugging in General Testing-Domain Dependent NHPP SRGM, *International Journal of Reliability, Quality and Safety Engineering*, 12(6), 493–505.
- Perry, W. (2005), Improving the Effectiveness of Software Controls - Gathering Feedback Information on Control, *The Journal of the Quality Assurance Institute*, January, 15-18.
- Pratt, J., A., Hauser, K., & Sugimoto, R., C. (2012), Cross-Disciplinary Communities or Knowledge Islands: Examining Business Disciplines, *Journal of Computer Information Systems*, Winter, 9-21

- Probst, G., Raub, S., & Romhardt, K., (2000), *Managing Knowledge. Building Blocks for Success*”, John Wiley & Sons, Polish version: *Zarządzanie wiedzą w organizacji*, Oficyna Ekonomiczna, Kraków 2002.
- Shang, S., S., C., & Lin, S., F. (2009), Understanding the Effectiveness of Capability Maturity Model Integration by Examining the Knowledge Management of Software Development Processes, *Total Quality Management*, 20(5), 509-521.
- Subramanian, A., M., & Soh, P.-H. (2009), "Contributing knowledge to knowledge repositories: dual role of inducement and opportunity factors", *Information Resources Management Journal*, 22(1), 45-62.
- Suma, V., & Gopalakrishnan Nair, T., R. (2008), Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels, *Proceedings of World Academy of Science, Engineering and Technology*, August 32, 288-292.
- Sun, Y., & Bhattacharjee, A. (2014), Looking Inside the “IT Black Box”: Technological Effects on IT Usage, *Journal of Computer Information Systems*, Winter, 1-15.
- Sureshchandar, G. S., & Leisten, R. (2006), Software Metrics for Enhanced Business Excellence: An Investigation of Research Issues from a Macro Perspective, *Total Quality Management*, 17(5), 609-622.
- Tsai, W.-H., Tsai, M.-H., Li, S.-T., & Lin, C. (2012), Harmonizing Firms’ Knowledge and Strategies with Organizational Capabilities, *Journal of Computer Information Systems*, Fall, 23-32
- Van Beveren J., (2002), *A model of knowledge acquisition that refocuses knowledge*, *Journal of Knowledge Management*, 6(1), 18-23.
- Wallace, D., & Kuhn, R. (2001), Failure Modes in Medical Device Software: An Analysis of 15 Years of Recall Data. *International Journal of Reliability, Quality and Safety Engineering*, 8(4), 351-371.
- West, J., E. (2007), Amendment to ISO 9001:2000-Quality Management Systems-Requirements and Revision of ISO 9004:2000-Quality Management Systems – Guidelines for Performance Improvement, *Quality Engineering*, 19, 137-142.
- Wiig, K., M. (2003), A Knowledge Model for Situation-handling. Knowledge Research Institute, INC. Texas, 1-29.

**Table 1.** Connecting QMS to the Hierarchy of Knowledge Description Levels for the Domain Basic Processes

	<b>Knowledge description level</b>	<b>Level description</b>	<b>Standard</b>
1	Region	Design and development	7.3
2	Segment	Design and development planning Design and development inputs Design and development outputs Design and development review Design and development verification Design and development validation Control of design and development changes Supervising changes in design and development	7.3.1 7.3.2 7.3.3 7.3.4 7.3.5 7.3.6 7.3.7 7.3.7
3	Knowledge element, solely for 7.3.1 knowledge segment	Planning of design and development Overview, verification, validation The scope of responsibility and competence Interface	7.3.1.1 7.3.1.2 7.3.1.3 7.3.1.4
4	Knowledge fragment for 7.3.1.4 knowledge element	Table 2	

**Table 2.** Knowledge Fragment for Knowledge Element (7.3.1.4)

<b>Knowledge fragment</b>	<b>Potential decisions localizing knowledge in the SDO</b>
7.3.1.4-1	Determining the suppliers' scope of responsibility for each part of the product with regard to planning, design and development
7.3.1.4-2	Determining the way in which technical information is circulated among all the parties
7.3.1.4-3	Determining other – apart from the client and organization – entities interested in activities relating to design and development
7.3.1.4-4	Determining other – apart from the client and organization – entities interested in activities relating to installation
7.3.1.4-5	Determining other – apart from the client and organization – entities interested in activities relating to use
7.3.1.4-6	Determining other – apart from the client and organization – entities interested in activities relating to maintenance
7.3.1.4-7	Determining other – apart from the client and organization – entities interested in activities relating to training