

Control Mechanisms for Assuring Better IS Quality

Marjan Pivka
 University of Maribor
 School of Business and Economics Maribor
 Razlagova 14, 62000 Maribor, Slovenia
 Tel: ++ 386 62 2290247, Fax: ++ 386 62 26 681
 Email: pivka@uni-mb.si

Keywords: software quality, IS manager, business

Edited by: Janez Grad

Received: August 28, 1997

Revised: July 17, 1998

Accepted: August 25, 1998

The software domain is faced with a number of quality assurance and process improvement models. Business managers are under pressure from many different kinds of assessments for their operations, products and services. Accounting departments are audited by financial auditors. What about Information Systems? Do we have a universal model on how to achieve required IS quality? This paper deals with the definition of IS quality and the influence of different control mechanisms on IS. The results of this empirical research are several. First of all, none of the control mechanisms are universal and applicable to all IS resources. Application of more than one of them could be redundant. Mutual recognition of results between them is required. IS managers are responsible to understand them and use them with all limitations on specific IS resources.

1 Introduction

The computer based Information System (IS) uses hardware, software, telecommunications and other forms of Information Technology (IT) to transform data resources into a variety of information products. Enterprises need and use those products in their business processes to achieve business objectives. IT resources need to be managed in order to provide such information products to the enterprise.

Typical resources of IS are:

dataware computer data bases and other data resources

software computer programs, applications,...

lifeware human resources

hardware computers, communications and other office technology

orgware organisation, procedures etc.

In business we are constantly under pressure to reduce all kinds of expenses on the one hand and to improve quality on the other. One of these expenses are those for Information Systems and there are always some logical questions to be asked: is it good enough to justify the expense? Do we get what we need? Shall we invest in new IS? Is our IS reliable? Are results accurate? Those and other questions can be answered in a discussion on the quality of IS. Because quality

is what we all expect from IS: cover of all functional requirements, reliability, needed results given on time, usable for all users, maintainable etc.

It would be too easy to suppose that IS quality could be achieved by assuring a high quality of IS resources. IS is too complex and it grows with organisation and needs different control mechanisms in its maturity process. The high quality of technical resources of an IS, such as software or hardware, is by no means any guarantee for its high quality implementation in an improperly organized enterprise! And vice versa!

Different methods and principles (our term is *control mechanisms*) are known for IS quality assurance. Some of them control the IS development process, others IS resources and some may be used to control the development process and the implementation process. The best known control mechanisms today are:

- quality system standards (ISO 9000 family standards);
- software products standards;
- software process assessment models such as BOOTSTRAP [Haase et al 1993], CMM [Paulk et. all 1993] [CMM v2.0], ISO/SPICE [Rout P. Terence 1995] and many others [SPC 1997];
- IS Auditing.

It is difficult to understand those, and perhaps other, aggressively marketed control mechanisms.

Many have asked themselves: ISO 9000, CMM, SPICE, IS Auditing, BOOTSTRAP - what shall I do? Why does our financial auditor request an IS Audit, if we use a certified software product, or have an ISO 9001 certificate in software development? What shall I do? Which mechanism or model can help me? What are their strengths, what are their weaknesses? The answer is not universal or easy.

The aim of this paper is to analyse the influence of the most well known control mechanisms for IS (not just software!) quality. The paper is organized as follows. The second section introduces a formal definition of IS quality. Sections three to six deal with quality system standards, software product standards, software process assessment models and IS Auditing. Each control mechanism is briefly introduced and its strengths and weaknesses on IS quality are discussed. The discussion on collaboration, competition or conflicts between those control mechanisms is presented in section 7. Finally conclusions are given in section 8.

2 The formal definition of IS quality

The quality is defined as *totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs* [ISO/IEC 8402:1995]. According to this definition of IS, the quality of IS in general is totality of needed or implied quality characteristics of dataware, software, lifeware, hardware, and orgware. Or speaking more generally, the quality of IS is totality of quality of IS resources. Therefore, if we want to judge whether a given information system is reliable or not, accurate or not, efficient or inefficient, etc. we shall:

- define quality model i.e. quality characteristics based on required and implied needs,
- measure or assess each characteristic,
- compare the measured or assessed characteristics with the specific requirements, and
- validate the results.

In the field of engineering these procedures are well defined and known as evaluation models. In software engineering, such a model is defined in [ISO/IEC 9126:1991] and in [ISO/IEC DIS 14598:1996]. Real implementation of this evaluation model requires the practical solution of some very serious problems:

1. The definition of IS quality model. A quality model in general is a structure or composition of all quality characteristics of an entity. Thus, for IS quality model the quality characteristics and

their sub-characteristics shall be defined for each IS entity.

2. Each quality characteristic shall be decomposed to a measurable or assessable level. The metrics assessment method for each characteristic has to be defined.
3. User's, legal and or professional requirements (i.e. stated and implied needs) and their significance for (on) IS quality shall be defined for each quality characteristic.

The quality of IS is of course not a simple sum of the quality of each IS resource. The quality of an IS Q is a function of the stakeholders defined (stated and - or implied) IS characteristics (DC_i), actual values of those characteristics (AC_i), and by stakeholders defined influence of each characteristic (p_i) on IS:

$$Q = f(DC_i, AC_i, p_i).$$

Function f , and its arguments DC_i , AC_i and p_i depends on management & operations maturity of the organisation, type of organisation, its environment etc. for which the IS is intended. They all define general requirements such as the type of IS (Management IS, Decision support systems, Executive Support Systems, ...or any combination thereof), and of course very specific requirements such as Inputs, Outputs, Interfaces, Security requirements, Services etc.

This definition formally demonstrates, that quality of IS can not be achieved by high quality of the IS resources (AC_i). It can be achieved only if requirements (f , DC_i and p_i) are demonstrated with actual resources (AC_i). For instance: the implementation of a high quality and complex software product in an badly organized organization will result in a badly organized IS!

3 Influence of Quality System standards on IS quality

The quality system is defined as *organisational structure, procedures, processes and resources to implement quality management* [ISO/IEC 8402:1995]. The most popular international quality system standards are ISO 9000 family standards. ISO 9001:1994 defines a model for quality assurance in design, development, production, installation and servicing. It defines a number of quality requirements structured in 20 clauses, from management responsibility to statistical techniques. ISO 9000-3:1997 [ISO 9000-3:1997] are guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software. British TickIT [TickIT 1998] certification schema assures a thorough compilation of

ISO 9001 with ISO 9000-3 guidelines in software development processes.

The result of ISO 9001 application in software development is the defined, controlled and managed process of development, supply, installation and maintenance of computer software.

The strengths of implemented ISO 9001 in a software development process are:

- The management of software development process is focused on internationally acknowledged quality requirements defined by ISO 9001 which is also measurable. As such, management can monitor and compare quality characteristics of the software development process (productivity, efficiency, number of bugs, user complaints etc) with their plans and with their competitors.
- It covers the processes in a software development, acquisition and implementation: from requirements definitions to planning activities, Software Life Cycle (SLC) activities, configuration management, hardware purchasing, software maintenance and training.
- It assures constant improvements in SLC, with Quality Assurance and Quality Control activities such as corrective and preventive actions.
- It improves co-operation between all parts in the SLC processes.
- Its international recognition (national and international certification schemes) has a strong impact on the software industry.

The weaknesses of implemented ISO 9001 in a software development process are:

- The quality system is limited to the software development processes. IS resources such as life-ware, orgware, hardware and other IS resources are not directly considered;
- Implementation of activities such as security management, application control and technology specific controls, depend on the maturity of the implemented quality system. This may vary from not implemented at all to fully implemented procedures;
- Implementation of the ISO 9001 standard does not give clear answers to questions concerning productivity, functionality, usability, reliability, cost effectiveness etc. of an IS.
- A controlled software development process is no guarantee for quality solutions to business problems. If the user defines a bad or inadequate requirement, this will be with high quality built in a software, but the end product will be of no use.

The above discussion consider only the software development process. An interesting situation is, where IT resources in an enterprise are under the umbrella of ISO 900x requirements, but other business processes are not. The information products are faced with problems like functionality, availability or usability in such a cases. The root causes are in the different maturity levels of business orgware. A Data Base Administrator can not design a robust and accurate data model of an enterprise if the enterprise has no business vision or policy statement, long and short term business plans, and also in the case when he or she may or can communicate only with people from third level management. The business objectives of an enterprise are not achieved only with the IT resources. IT resources are the support to other processes and ideally, all of them must be on the same maturity level to achieve expected business results.

The implementation of the ISO 9001 standard in the software development process has no essential influence on a very important IS source, orgware. This means that a quality of IS can not be assured only by implementation of the ISO 9001 standard in the software development process, but it is also necessary to consider the maturity of the organisational and management level of the company. In other words, the impact of ISO 9001 in software development and on IS depend on the maturity of the environment for which the software development is intended.

However, it is to be expected that the gap between IT processes and business processes will be reduced in time if the quality system is implemented only in one of them. A quality system requires internal audits and corrective and preventive actions which assure improvements and growth of all involved.

4 Influence of software product standards on IS quality

There are several hundred software standards. Most of them are National or Multinational such as ANSI (American National Standard Institute), BSI (British Standard Institute) and DIN (Deutsches Institute für Normung) standards, and professional standards such as IEEE Standards, Defence standards etc. The majority of them deal with the results of software activities and tasks such as Management, Quality Assurance, Configuration Management, Safety, Design, Requirement Specification, Coding, Verification & Validation etc. and are mainly considered in the process of software development (discussed in sections 2,4, and 5). The applicability of them and their influence on IS quality therefore depends on the scope of the standard within the software development process.

Only a few of the software product standards deal with software products for end users or software pack-

ages. They are: ISO/IEC 12119 1995: *Information Technology - Software packages - Quality requirements and testing*, ISO/IEC 9126 1991: *Information Technology - Software product evaluation - Quality characteristics and guidelines for their use* and ISO 14598:1996 - Part 1 to Part 6: *Information Technology, Software product evaluation*.

The international standard ISO/IEC 12119 is applicable to software packages like accounting, payroll, data base programs etc. Sets of quality requirements based on this standard are: requirements on product description, documentation, programs and data, and testing procedure.

The strengths of software product standards are:

- Conformity according to those standards provide confidence that the product actually does what it claims to.
- They can be used in national or international software packages certification schemes.
- They can be used as a marketing advantage for off the shelf products;
- They can have a substantial impact in a software acquisition process.

The weaknesses of those standards generally deal with their scope and their importance in the IS. Existing software product standards cover only some parts of the software design process or software products. Its influence on IS quality is limited to the importance of the considered subject of software quality. It is obvious that safety standards (implemented in a software development process!) have a supreme influence on IS quality in safety critical systems, such as Nuclear Power Plant. And vice versa: a text processor package with a certificate of conformity with ISO/IEC 12119, has only little influence on IS quality.

5 Influence of software process assessment models on IS quality

It is very important to recognise, that any software process improvement program needs a sound understanding of the current status of the software development process. Software process assessment is the most common method used to achieve this understanding. Process assessment is defined as *The disciplined examination of the process used by an organisation against a set of criteria to determine the capability of those processes to perform within quality, cost and schedule goals. The aim is to characterise current practice, identifying strengths and weaknesses*

and the ability of the process to control or avoid significant causes of poor quality, cost and schedule performance. [ISO/IEC JTC1/SC7 1992]. This definition is also applicable to software process assessment.

The most popular approaches for software process assessment are the Software Engineering Institute's (USA) CMM - Capability Maturity Model [Paulk et. al 1993] [Paulk M.C. 1995] [CMM v2.0], ISO/SPICE (Software Process Improvement and Capability Determination) project [ISO 15504 (SPICE) PDTR Draft 1996], and BOOTSTRAP (European developed assessment method [Haase et al 1993]). Some of the others are: Capers-Jones software measurement model [Jones 1991] and Model-based Process Assessment [McGowan et al 1993]. SLC is also subject of standardisation: IEEE standard for software life cycle processes [IEEE 1988] and ISO/IEC12207: 1995 Information Technology - Software life cycle processes (SLC). Many other models and standards can be found in the Software Productivity Consortium WEB server [SPC 1997].

The CMM model provides a conceptual structure for improving the management and development of a software process in a disciplined and consistent way. The CMM model divides the software process into five maturity levels which highlight the primary process changes made at each level: Initial or basic level (*ad hoc* process), Repeatable (basic project management is established), Defined (process is documented and standardised), Managed (process and product measurements are established) and Optimised. Each level comprises of a set of process goals that, when satisfied, stabilise an important part of the software process. This in turn results in an increase of the software process capability. Each maturity level is composed of a number of key process areas. These key process areas are a set of activities that, when implemented, achieve a set of goals important for enhancing the process capability.

A European software process assessment and improvement method - BOOTSTRAP - has been developed in an ESPRIT (1990 - 1993) project. BOOTSTRAP has been built on the basis of the CMM and ISO 9000 series of standards. The basic concept underlying BOOTSTRAP requires first fulfilment of basic organisational requirements, such as process control, project management and risk management before any changes in methods and technology are made to improve the software process. An organisation and its processes are assessed with respect to organisation, methodology and technology. The result of BOOTSTRAP assessment is a capability profile showing maturity of an organisation against an ideal level of maturity, comparison with ISO 9001 requirements and recommendations for appropriate actions for further improvements.

ISO/SPICE is a project of the international Committee on Software Engineering Standards ISO/IEC JTC1/SC7. It synthesises above mentioned models and standards (CMM, Bootstrap, Trillium, ISO 9001, ISO 12207 and others). SPICE embodies a sophisticated model for software process management drawn from the world-wide experience of large and small companies. The architecture of the SPICE process assessment defines a two-dimensional view of software process capability: the process categories and capability level. Process categories are: Customer-supplier process category, Engineering process category, Project process category, Support process category and Organisation process category. Each category is a set of processes addressing the same general area of business. The result of SPICE assessment is a "software process profile" defining a capability level for a considered software process category. This model will have a significant influence on software domain in the near future: very probably as "the facto standard".

The strengths of software process assessment models are that:

- The Software Life Cycle processes are dealt with in full.
- They give a clear profile of the current capability of the software development and maintenance processes. This profile corresponds to the maturity level of the software development and maintenance process. Maturity levels or maturity profiles identifies current capabilities of the process and identify process areas for further improvements.
- They have a important influence on the software community.
- The results of an assessment can be used in a benchmarking process.
- The results can be used in national or international procurement activities.
- They can be used for self assessment, as a starting point into a quality improvement program.

The weaknesses of those models are:

- They are limited to the Software Life Cycle and do not consider other IS/IT resources such as orgware, peopleware, hardware, telecommunications etc.
- They are not known outside of the software community (to the IT users).
- They are neither national or international standards.

Companies where those models are applicable are software houses and Electronic Data Processing departments or software development departments within enterprises. They can be used as a self assessment tool for improvements plans or implemented by a third party as an independent assessment, if stakeholders require such an assessment.

6 IS Auditing

The EDP Auditor Foundation, Inc. (EDPAF) developed General standards for information system auditing [Dykman A.C.] and Control objectives as a model for IS audit procedure. By general standards for information systems auditing [IS Audit], the Information System Auditing is defined as *any audit that encompasses the review and evaluation of all aspects (or any portion) of automated information processing systems, including related non automated processes, and the interfaces between them.* Those aspects, defined by EDP auditor Control objectives [Dykman A.C.] are:

- Management control
- Information system development, acquisition, and maintenance
- Information system operations controls
- Application controls
- Database supported information system controls
- Distributed data processing and network operations controls
- Electronic data interchange controls
- Service bureau operations controls
- Micro computer controls
- Local area network controls
- Expert system controls and
- Joint application design controls.

Each general control is divided in to controllable and manageable units. From the definition of the IS, the IS audit definition by EDPAF, and the described control objectives it may be concluded that EDP audit procedures deals with all aspects of an IS. The result of the IS audit procedure is a set of documented facts obtained with interviews and questionnaires by a certified auditor on an audited IS entity.

The strengths of the IS auditing assessment model are:

- any IS/IT entity can be audited;
- it is a strong management tool;

- special considerations are pointed on technical aspects such as Data Bases, LANs, Micro Computer Control, application controls, IT/IS risk evaluation and data security etc.;
- it is usually used in collaboration with financial auditing;
- it can be used as a basis for an improvement program;

The weaknesses of this model are:

- it can not be used as a tool to find out the current maturity level of a software development process or to compare a project profile;
- it can not be used in an competition attaining work / business
- it is neither a national or international standard;
- it can not be used as a self assessment model;

This model is applicable mostly to the auditing of EDP departments and other IS resources within enterprises [Pivka M. 1998]. Owners of IS auditing are usually company's management and accounting auditors. IS auditing is generally not applicable to software houses. Detail informations on control objectives for information and related technology are in [CobiT 98] which is a registered trade mark of ISACA.

7 IS control mechanisms: collaboration, competition or conflicts?

Information Technology managers, software development managers, business managers and users are faced with aggressively marketed control mechanisms. The following questions are interesting from the IS management point of view:

1. Which model to choose?
2. Which IS resources are controlled by those mechanisms?
3. Are they redundant?
4. Do they compete?
5. Do they collaborate?

The answer to those and other questions on IS control mechanisms are not easy and universal. The following paragraphs describe the most general characteristics of control mechanisms and their influence on IS resources.

In table 1, the ranking of the influence of control mechanisms on some IS resources is defined. It is obvious, that assessment models have limited or nil influence on technology resources such as communications. But of course, they have a strong influence on the software development process.

Table 2 represents which model to choose for some most interesting business requirements.

Quality systems based on ISO 9001, BOOTSTRAP, SPICE, CMM and other assessment models, are **management tools** for improving the software development, maintenance and implementation process. They have an important influence on the software environment in defining their maturity level and in helping them to find and define the key management procedures to improve the software process. Assessment teams (first, second or third party teams) use BOOTSTRAP, SPICE or the CMM model to identify the maturity level of the software process. Models are not used in national or international certification schemes. On the other hand, the ISO 9001 certificate confirms at an international level that the software process is in compliance with internationally accepted quality requirements. The influence on IS of those models is therefore limited to the scope of the model! We may conclude, that there is some competition and conflicts between them, but also a possibility for collaboration. For instance: self assessment with a CMM model can be a good starting point for an improvement program with the aim of attaining an ISO 9001 certificate.

The influence of **software product standards** on IS quality is limited to the importance of the considered software package or the scope of the standard. Those standards shall therefore be recognised as helpful and useful in assessing aspects such as risk management, application control, data security, or any other IS/IT entity, where such standards exist and are implemented.

The **EDPAA IS audit model** is a tool for general management to evaluate the efficiency, security, productivity etc. of implemented IS/IT, or part of it, in a company. IS auditing does not deal with the natural growth of software processes as proposed by the BOOTSTRAP, SPICE, CMM, or with quality systems as defined in ISO 9000 family standards. IS audits are most usually ordered by top management or accounting auditors. There are some gaps between IS auditing and other models, which are for sure possibilities for collaboration. At least the following aspects are a matter of collaboration: application controls, risk management and IT assessment.

There is also some overlapping (and therefore, conflicts and competition) between those models, especially between assessment models. BOOTSTRAP, SPICE and CMM models overlapped each other and to conduct more than one of them in practice, is redundant. **Which of them to chose is in our opin-**

	ISO 9001	CMM, SPICE BOOTSTRAP	Product standards	IS Auditing
People	Strong	Strong	none	Medium
Application system	Strong	Medium	medium, depends on the standard and the application system	Strong
Software development	Strong	Strong	medium, depends on the standard and applied SLC	Medium
Technology of IS	Medium	Medium	none	Strong
Facilities	Medium	None	none	Strong
Data security	Depends on the maturity of the company: from medium to strong	depends on the maturity of the company: from medium to strong	standard dependable	Strong
Risk management	Depends on the maturity of the company: from medium to strong	depends on the maturity of the company: from medium to strong	standard dependable	Strong
Communications	Medium	Medium	standard dependable	Strong

Table 1: Control mechanisms and their influence on IS

Requirement: To assess specific IS/IT sources concerned productivity, security, usability, ...	Requirement: To assess software process as the basic for software improvement programme.	Requirement: To buy (or produce) SW product for market with legal or specific requirements	Requirement: To improve software quality and to get market Advantage
Possible solutions: IS auditing, combined with assessment methods if necessary for software process. Typical stakeholders: top management, financial auditors.	Possible solutions: Assessment methods: BOOTSTRAP, CMM, SPICE, ISO 9001. Depends on added value from assessor company. Typical stakeholders: SW managers, user requirement.	Possible solutions: Software product standards or ISO 9001. Typical stakeholders: legal requirements, market or contractor's demands.	Possible solutions: ISO 900x certification with one of the assessments methods. Usually influenced by market demands or by management.

Table 2: which model to chose.

ion the matter of added value, given from the method and assessor company (third party assessment). It is generally accepted that the software process which is certified to be in compliance with ISO 9001 is on the third capability level based on the CMM model. Overlapping between IS Auditing and other control mechanisms is obvious only with the software development and maintenance process and as that, SLC processes are an aspect of possible conflicts and competitions.

The responsibility for proper decisions between different models is unfortunately on the stakeholders side. **In situations where IS auditors, auditors for quality systems (ISO 9000 family) and software process assessors are hired, or some assessment results exist from the past, we recommend that company management demands from all parties collaboration and a mutual recognition of their results.** This shall have a substantial influence on the costs and added value of the company. Otherwise, a lot of people in the company will be interviewed several times with similar questions on the same subjects! Table 2 below shows an example of which model to choose for some business targets or strategies.

8 Conclusions

In this paper only the most popular and well defined control mechanisms to achieve better IS quality are briefly presented. Those control mechanisms are: ISO 9001 quality system standards, software process assessment models (CMM, BOOTSTRAP, SPICE), software product standards and IS Auditing.

The answer to the question which of them to choose is therefore not easy and it depends on the perspective of the stakeholder: enterprise management, IS management, buyer of software package, contractor for software or software services etc. Once the scope and required goals and expectations are defined and strengths and weakness of available control mechanisms understood then the right choice is not so difficult any more. This can be derived from the descriptions above. It is also obvious, that some competition, conflicts and possibilities for collaboration exist between the considered models.

It is the management's role and responsibility to understand those models and to avoid different audits on the same IS resource.

Acknowledgements

I would like to express my thanks to Gilles MOTET (DGEI/INSA Toulouse, France), Régis FLEURQUIN (IUT Informatique Vannes, France) and Dr. Walter WINTERSTEIGER (Management & Informatik

Dornbirn, Austria) for their important and valuable comments on the paper.

References

- [Haase et al 1993] V. Haase, R. Messnarz, G.Koch, H.J.Kugler, P. Decrinis: BOOTSTRAP: Fine-tuning process assesment, IEEE Software, July, 1994 pp 25-35.
- [Dykman A.C.] Dr. Charlene A. Dykman (editor): Control objectives. Controls in an Information systems Environment: Objectives, Guidelines, and Audit Procedures. EDP Auditors Foundation, Inc. Illinois, USA, 1992.
- [CMM v2.0] Draft version of CMM v 2.0. Internet web site <http://www.sei.cmu.edu>
- [CobiT 98] Control Objectives for Information & related technology. ISACA, Illinois, USA, 1998
- [IEEE 1988] Institute of Electrical and Electronics Engineers Inc., Standard of Software Life Cycle Processes, P1074/D2.1 Dec. 1988.
- [IS Audit] IS Audit & Control Journal: General standards for IS auditing. Vol. I. 1994, pp 60 - 66.
- [ISO 9000-3:1997] Quality Management and Quality Assurance Standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software.
- [ISO/IEC JTC1/SC7 1992] The Need and Requirements for a Software Process Assessment Standard. Study Report Issue 2.0 JTC1/SC7 N944R.
- [ISO/IEC 8402:1995] Quality management and quality assurance - Vocabulary.
- [ISO/IEC 9126:1991] Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.
- [ISO/IEC 12119:1995] Information Technology - Software packages - Quality requirements and testing.
- [ISO/IEC 12207:1995] Information technology - Software life cycle processes
- [ISO/IEC DIS 14598:1996] Information technology - Software product evaluation. Part 1: General overview. Part 5: Process for evaluators.
- [ISO 15504 (SPICE) PDTR Draft 1996] Software Process Improvement and Capability dEtermination (Software Process Assessment Standard). SPICE web site: <http://www-sqi.cit.gu.edu.au/spice/suite.shtml>

- [Jones 1991] T. Capers Jones, Applied Software Measurement, McGraw Hill, 1991.
- [McGowan et al 1993] C.L.McGowan and S.A.Bohner, Model Based Process Assessment, Proc. Int. Conf. Software Engineering IEEE CS Press, Los Alamitos, 1993.
- [O'Brien J. A. 1994] Introduction to Information Systems. IRWIN 1994.
- [Paulk et. al 1993] M. Paulk, Bill Curtis, M.B. Chriss&C Webber, Capability Maturity Model, Version 1.1. IEEE Software, July 1993, pp18-27.
- [Paulk M.C. 1995] Paulk M.C. How ISO 9001 Compares with the CMM. IEEE Software, January 1995, pp. 74-83.
- [Pivka M. 1998] Pivka Marjan. A Comparison of Control Mechanisms to Help Achieve Better IS Quality. ISAudit & Control Journal Volume II 1998.
- [Rout P. Terence 1995] SPICE: A Framework for Software Process Assessment. Software Process Improvement and Practice. August 1995.
- [SPC 1997] Software Productivity Consortium: <http://www.software.org/quagmire>.
- [TickIT 1998] The TickIT Guide. Issue 4.0. DISC TickIT Office. London 1998.