◆ Application Papers

A Study of the Implementation Guidance to ISO 9001:2000 in the Computer Software Industry

Lee, Byung Yong *
Jung, Soo Il**

Abstract

The International Organization for Standardization(ISO) is a worldwide federation of national standards bodies. Through ISO Technical Committees(TC), various International Standards are being carried out. Each member body interested in a subject for which a TC has been established has the right to be represented on that committee. ISO collaborates closely with the International Electro-technical Commission(IEC) on all matters of electro-technical standardization.

ISO established the ISO 9000 Family standard in 1987, and International Standard ISO 9000-3 was worked by ISO/TC 176, Quality management and quality assurance, Subcommittee 2(SC 2), Quality systems, in accordance with the ISO/IEC Directives, Part 3: 1997 Rules for the structure and drafting of International Standards, Many organizations have applied the ISO 9000-3 for their quality system standard in the software sector.

That means that ISO 9000-3: 1991 and ISO 9000-3: 1997 have been used successfully by the software industry as the internationally accepted interpretation of ISO 9001 for the development and maintenance of computer software. Additionally ISO 9000-3: 1997 involved how the software life cycle processes defined in ISO/IEC 12207: 1995, Information Technology - Software Life Cycle Processes related to the requirements of ISO 9001: 1994.

After having performed full reviews of the WD3, CD1, CD2 and DIS drafts of the future ISO 9001:2000, this document will partly replace the part of ISO 9000-3: 1997 for measurement, analysis and improvement of quality management system in computer software industry, as an interpretation for organizations and certification bodies, which will be withdrawn when ISO 9001: 1994 is replaced by ISO 9001: 2000,.

---

* Department of Industrial Engineering, Graduate School Inha University

** Professor, Department of Industrial Engineering Inha University

I. Introduction

ISO/DIS 9001:2000 defines the quality management system requirements for the organizations which develop computer software and provide its related services to the customer., This document will support to establish and to audit the quality management system for such organizations which are going to trade internationally, and provide internationally agreed guidance on the interpretation of the future ISO 9001:2000 for such organizations.

ISO/DIS 9001:2000 consists of 4 major chapters, those are *"management responsibility", "resource management", "product realization", "measurement, analysis and improvement"*. An organization is only allowed to exclude those quality management system requirements in *"product realization"* of the standard that does not affect its ability to offer products that conform to customer requirements. This document deals only with *"measurement, analysis and improvement"* to interpret those requirements of ISO/DIS 9001:2000, and that aids to improve and audit continually an organization's management quality system ability to offer conforming software products. This Document does not identify any requirements that do not apply to computer software.

II. Process model

ISO/DIS 9001:2000 contains a 'process model' depicting how the four primary requirements of the standard are related to each other and to customers. Table 1 below shows the relation between the requirements of draft ISO/DIS 9001: 2000 and the processes of ISO/IEC 12207:1995.

draft ISO/DIS 9001:2000 Requirement		**ISO/IEC 12207:1995 Process**
4 Quality management system	4.1 General requirements	
	4.2 General documentation requirements	
5 Management responsibility	5.1 Management commitment	7.1 Management
	5.2 Customer focus	5.1 Acquisition 5.2 Supply
	5.3 Quality policy	
	5.4 Planning	
	5.5 Administration	
	5.6 Management review	
6 Resource management	6.1 Provision of resources	
	6.2 Human Resources	7.4 Training
	6.3 Facilities	7.2 Infrastructure
	6.4 Work environment	
7 Product realization	7.1 Planning of realization processes	5.3 Development

draft ISO/DIS 9001:2000 Requirement		ISO/IEC 12207:1995 Process
		5.4 Operation 5.5 Maintenance 6.1 Documentation 6.2 Configuration Management 6.3 Quality Assurance 6.4 Verification 6.5 Validation 6.6 Joint review
	7.2 Customer- related processes	5.1 Acquisition 5.2 Supply
	7.3 Design and/or development	5.3 Development 6.2 Configuration Management 6.4 Verification 6.5 Validation 6.6 Joint review
	7.4 Purchasing	5.1 Acquisition
	7.5 Production and service operations	5.4 Operation 5.5 Maintenance 6.2 Configuration Management
	7.6 Control of measuring and monitoring devices	6.2 Configuration Management
8 Measurement, analysis and improvement	8.1 Planning	6.4 Verification 6.5 Validation 6.6 Joint review 6.7 Audit 7.3 Improvement
	8.2 Measurement and monitoring	6.7 Audit 7.3 Improvement
	8.3 Control of nonconformity	6.2 Configuration Management
	8.4 Analysis of data	7.3 Improvement
	8.5 Improvement	6.2 Configuration Management 7.3 Improvement

Table 1: Cross-references between draft ISO/DIS 9001:2000 and ISO/IEC 12207:1995.

III. Measurement, analysis and improvement

1. Planning

The organization shall define, plan and implement the measurement and monitoring activities needed to assure conformity and achieve improvement. This shall include the determination of the need for, and use of, applicable methodologies including statistical techniques.

The measurement, analysis and improvement processes, and the method/frequency of measurements and the requirements for quality records, should be specified as part of the quality planning . To ensure that the quality management system meet to customer requirements, a software developing organization should:

1) measure the conformity of products and/or services to customer requirements by means of review, verification and validation processes (see ISO/IEC 12207 :1995 clauses 6.4, 6.5 and 6.6);

2) measure customer satisfaction with products and/or services to check that the quality management system is achieving its quality objectives (see Ⅲ.2.1);

3) conduct internal audits to check that the quality management system procedures are followed (see Ⅲ.2.2 and ISO/IEC 12207 :1995 clause 6.7);

4) conduct software process assessments to identify process improvements that increase the degree of conformity to requirements (see ISO/IEC 12207:1995 clause 7.3 and ISO/IEC TR 15504:1998).

The term "metrics" mean measurable characteristics of products and/or services in software engineering.

Examples of metrics to software product are testability; usability; reliability; maintainability and availability.

Examples of metrics to software process are process maturity, and number and type of defects in process outputs, defect removal efficiency, and milestone slippage.

The effectiveness of metrics should be evaluated by estimating their capability:

1) to predict customer satisfaction with the product and/or service, or;

2) to report the probable number of defects present in the product, or;

3) to report the degree of progress towards cost, schedule and quality objectives.

When the size and complexity of a software product implies that complete verification that a software product meets its requirements is not economically practical, statistical techniques should be used to estimate the:

1) the proportion of software product parts that need to be processed by each verification activity (e.g. the proportion of lines of code that should be covered in testing, the proportion of code that should be inspected);

2) the number of outstanding defects in the software product after each stage of verification (e.g. the number of defects per hundred lines of code after testing).

Statistical techniques may also be used to estimate the reliability and maintainability of the software product from data collected during development, operations and maintenance (e.g. mean time between failures, mean time to correct the failures).

2. Measurement and monitoring

2.1 Customer satisfaction

The organization shall monitor information on customer satisfaction and/or dissatisfaction as one of the measurements of performance of the quality management system. The methodologies for obtaining and using this information shall be determined.

A software development and maintenance organization should evaluate customer satisfaction by analyzing:

1) all nonconformities detected by customers during development and maintenance;

2) all complaints from customers during development and after delivery;

3) all survey results of satisfaction with the software product and/or service;

4) costs and benefits of the software product and/or service to customers.

A software development and maintenance organization should establish an improvement process (see ISO/IEC 12207:1995 clause 7.3) for establishing, internal auditing, measuring and improving a software life cycle process. The management review activity should be combined with the improvement activity. Software process assessment activities should be based upon the requirements and guidelines in ISO/IEC TR 15504:1998.

2.2 Internal audit

The organization shall conduct periodic internal audits to determine whether the quality management system: a) conforms to the requirements of this International Standard; b) has been effectively implemented and maintained. The organization shall plan the audit program taking into consideration the status and importance of the activities and areas to be audited as well as the results of previous audits. The audit scope, frequency and methodologies shall be defined. Audits shall be conducted by personnel other than those who perform the activity being audited. A documented procedure shall include the responsibilities and requirements for conducting audits, ensuring their independence, recording results and reporting to management. Management shall take timely corrective action on deficiencies found during the audit. Follow-up actions shall include the verification of the implementation of corrective action, and the reporting of verification results. Note: See ISO 10011 for guidance

Internal audits should be periodically performed during development and before the release of a software product with the quality objective of checking that:

1) the capabilities of the software product are consistent with its requirements (i.e. a functional audit);

2) the contents of the software product are consistent with its documentation (i.e. a physical audit).

A functional audit may be performed at module test stage, then system test stage, and then integrated test stage.

Audits performed to identify potential opportunities for improvement of the processes of a software

developing organization should be called "software process assessments", and should be based upon ISO/IEC TR 15504:1998.

When software development and maintenance work is organized into projects, the internal audit schedule should define a selection of projects to be audited. To ensure that the whole quality management system is audited, projects at different stages of their life cycle should be included in the internal audit schedule, or the same project at different stages of the life cycle be included,or both.

2.3 Measurement and monitoring of processes

The organization shall apply suitable methods for measurement and monitoring of those realization processes necessary to meet customer requirements. These methods shall confirm the continuing ability of each process to satisfy its intended purpose.

To measure key processes necessary to meet customer requirements and demonstrate the processes' continuing ability to satisfy its intended purpose, a software developing organization should:

1) establish and operate an improvement process (see ISO/IEC 12207:1995 clause 7.3);

2) perform internal audits (see Ⅲ.2.2);

3) perform software process assessments (see ISO/IEC TR 15504:1998).

2.4 Measurement and monitoring of product

The organization shall measure and monitor the characteristics of the product to verify that requirements for the product are met. This shall be carried out at appropriate stages of the product realization process. Evidence of conformity with the acceptance criteria shall be documented. Records shall indicate the authority responsible for release of product. Product release and service delivery shall not proceed until all the specified activities have been satisfactorily completed, unless otherwise approved by the customer.

A software developing organization should apply the following methods for measuring of the characteristics of the product and/or service to verify that requirements for the product and/or service are met:

1) design and development review ;

2) design and development verification ;

3) design and development validation ;

4) suitable activities after delivery ;

5) customer satisfaction monitoring (see Ⅲ.2.1).

3. Control of nonconformity

The organization shall ensure that product which does not conform to requirements is identified and controlled to prevent unintended use or delivery. These activities shall be defined in a documented procedure. Nonconforming product shall be corrected and subject to re-verification after correction to demonstrate conformity. When nonconforming product is detected after delivery or use has started, the organization shall take appropriate action regarding the consequences of the nonconformity. It will often be required that the proposed rectification of nonconforming product be reported for concession to the customer, the end-user, regulatory body or other body.

A software developing organization should control nonconforming product and/or service by:

1) ensuring that documents include their version and approval information;

2) identifying the inspection and test status of software items either by segregating them according to their inspection and test status, or by maintaining records of software items that identify their inspection and test status, or both;

3) maintaining records that identify the nonconforming product and the nature of the non-conformity.

The configuration management system of a software developing organization should include configuration control procedures for identifying, recording and reviewing the nature and extent of the nonconformity encountered (see ISO/IEC 12207:1995 clause 6.2.3).

The configuration control procedures of a software developing organization should include or refer to procedures for the review of nonconformities (see ISO/IEC 12207:1995 clause 6.2.3). Considerations to the review should include:

1) a diagnosis of the cause of the nonconformity, if it exists;

2) an importance of the nonconformity and the consequences of its continued existence;

3) an evaluation of the impact of the proposed change to the software;

4) a re-test records that will verify that the nonconformity has been removed;

5) a "regression" tests records that will verify that new nonconformity has been recurred.

Disposition of nonconforming software products may be achieved by:

1) repair or rework (i.e. to fix defects) to meet the requirement;

2) acceptance with or without repair by concession;

3) treatment as a conforming product after the amendment of requirements;

4) reject.

Repair or rework to software modules in a new version of the module and any product according to configuration control.

4. Analysis of data

The organization shall correct and analyze appropriate data to determine the suitability and effectiveness of the quality management system and to identify improvements that can be made. This includes data generated by measuring and monitoring activities and other relevant sources. The organization shall analyze this data to provide information on: a) customer satisfaction and/or dissatisfaction; b) conformance to customer requirements; c) characteristics of processes, products and their trends; d) suppliers.

The quality management system of a software developing organization should enable it to achieve the quality, cost and schedule targets for the development and maintenance of the software product and/or service that it delivers.

A software developing organization should analyze the followings for improving the effectiveness of the quality management system:

1) the number, type and severity of customer complaints;

2) the number, type and severity of nonconformities reported by the customer after delivery;

3) the number, type and severity of nonconformities detected at each stage of development and maintenance;

4) the time difference between planned and actual events, such as the achievement of project milestones.

5) the testability, usability, reliability, maintainability and availability of the software products and services as metrics;

6) the costs incurred by the customer as a result of each nonconformity;

7) the costs incurred by the organization for the analysis of the nonconformity and any rework;

8) the costs of each stage of the development and maintenance process;

The procedures for reporting the progress of projects and the history of projects should include instructions for the analysis of data for improving the performance of the current project, other projects, and the quality management system.

The establishment of an improvement process (see ISO/IEC 12207:1995 clause 7.3) ensures that procedures for the analysis of applicable data exist.

5. Improvement

5.1 Planning for continual improvement

The organization shall plan and manage the processes necessary for the continual improvement of the quality management system. The organization shall facilitate the continual improvement of the quality management system through the use of the quality policy, objectives, audit results, analysis of data, corrective and preventive action and management review.

The establishment of an improvement process (see ISO/IEC 12207:1995 clause 7.3) ensures that procedures for the continual improvement of the quality management system of a software developing organization exist.

5.2 Corrective action

The organization shall take corrective action to eliminate the causes of nonconformities in order to prevent recurrence. Corrective action shall be appropriate to the impact of the problems encountered. a) identifying nonconformities (including customer complaints); b) determining the causes of nonconformity; c) evaluating the need for actions to ensure that nonconformities do not recur; d) determining and implementing the corrective action needed; e) recording results of action taken; f) reviewing of corrective action taken.

The configuration control procedures (see ISO/IEC 12207:1995 clause 6.2.3) of a software developing organization should define how to:

1) identify nonconformities;

2) evaluate the need for actions to ensure that nonconformities do not recur (see III.3);

3) record that the results of actions taken;

4) review that corrective action is effective and recorded.

The software development or maintenance procedures (see ISO/IEC 12207:1995 clauses 5.3 and 5.5) should define how to:

1) analyze the causes of nonconformities;

2) implement any actions to ensure that nonconformities do not recur.

The identification of a nonconformity should include a description of the nonconformity in sufficient detail to enable it to be reproduced. The analysis of the cause of a software nonconformity is normally performed by reviewing the nonconformity in a test environment.

The software development, maintenance and configuration control procedures may be part of the quality

plan for the software product and/or service .

5.3 Preventive action

The organization shall identify preventive action to eliminate the causes of potential nonconformities to prevent occurrence. Preventive actions shall be appropriate to the impact of the potential problems. The documented procedure for preventive action shall define requirements for: a) identifying potential nonconformities and their causes; b) determining and ensuring the implementation of preventive action needed; c) recording results of action taken; d) reviewing of preventive action taken.

Analysis of the root causes of nonconformities may provide input to preventive action. The measures taken to reverse unfavourable trends in metric levels may be considered as preventive actions.

To prevent nonconformities in software product and/or service, a software developing organization should:

1) establish suitable development and maintenance processes that minimize the number and severity of any defects that could be introduced during development and maintenance;

2) ensure that review, verification and validation processes are performed before the outputs of each development or maintenance activity are input to the next stage;

3) ensure that the configuration control activity, when processing nonconformities, considers preventive actions need to;

4) remove defects in development or maintenance processes that result in the non-conformities in the product and/or service;

5) remove other defects in the product and/or service that have been detected during problem diagnosis of the nonconformity.

The procedures for configuration control, development and maintenance should meet the requirements for a procedure for preventive action in the same way that they meet the requirements for a procedure for corrective action (see Ⅲ.5.2).

Ⅳ. Conclusion

The Quality Management System requirements of the future ISO 9001:2000 are intended to be applied for the quality management of all types of products, including computer software. Existing International Standards, ISO/IEC 12207:1995 *Information Technology – Software Life Cycle Processes*, that defines the processes that should be used during the life cycle of software, and ISO/IEC 9126:1991, *Information*

technology – Software product evaluation – Quality characteristics and guidelines for there use , and this document are to define how software life cycle processes can be used to implement the Quality Management requirements of the future ISO 9001:2000 for the organization.

For that purpose this document should be completed with interpretation of the rest parts of the future ISO 9001:2000. And the future ISO 9004:2000 will contain guidance for the performance improvement of Quality Management Systems. Then an existing Technical Report, ISO/IEC TR 15504:1998, *Information Technology – Software Process Assessment*, that defines how to perform software process assessment, and this complete document are used to define how software process assessments can be used to achieve performance improvement.

References

1. ISO/DIS 9000:2000, Quality Management Systems - Requirements
2. ISO/TC 176/SC 2/N 376:1997, Quality management principles and guidelines on there application.
3. 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.
4. ISO/TC 176/SC 2/N 442: 1999, Proposal for the revision of ISO 9000-3:1997 with ISO/CD2 9001:2000 .
5. ISO 10007:1995, Quality management - Guidelines for configuration management.
6. ISO 10012-1:1992, Quality assurance requirements for measuring equipment – Part 1: Metrological confirmation system for measuring equipment.
7. ISO/IEC 9126:1991, Information technology – Software product evaluation – Quality characteristics and guidelines for there use.
8. ISO/IEC 12119:1994, Information technology – Software packages – Quality requirements and testing.
9. ISO/IEC 12207:1995, Information technology – Software life cycle processes.
10. ISO/IEC TR 15504:1998, Information technology – Software process assessment
11. Guidelines for assessment and implementation of quality systems in the information technology sector, 1997. N.V. KEMA