# SOFTWARE QUALITY MANAGEMENT THROUGH IMPLEMENTATION OF SOFTWARE STANDARDS

**S. Grceva, Z. Zdravev**

Faculty for Education "Goce Delcev", University of "Sts. Cyril and Metodius"
Goce Delcev bb, Stip
E-mail: {solza_grceva, zdravev}@yahoo.com

**Abstract:** Many of the tools of Quality Management are based around the concept of a "process". The simplest example is a manufacturing production line. Non-manufacturing companies can also be seen as collection of different processes. A software development company has its software development process with user requirements on the one end and the finished software product on the other. As the manufacturing industries have to comply with certain production standards, the large software companies and government agencies are increasingly using Quality Assurance (QA) Certification as their basic strategy for managing competition and supplier relationships. The key factor is "reliability". It is assurance that the quality achieved today can be supplied well into the future. The reliability can only be demonstrated by pointing to the systems (the policies, methods, processes and procedures) used to build up quality products and making a commitment to maintaining these. This is what QA Certification and standards are really about.

**Keywords:** Software Standards, Quality Management, Quality Assurance Certification

## 1 Introduction

The goal of this paper is to provide a practical guide on how to achieve software process improvement and gain higher quality products through use of the existing standards and models. Also to give an understanding of what standards and models are and what they can deliver; to show how standards can be selected and tailored to meet the very different needs of the developers and the customers; finally, to examine the relationship between quality and software process improvement strategies and standards.

The paper addresses, highlights and compares some points of interest of the most important standards and models like: ISO/IEC/IEEE/EIA 12207, ISO 9001, the IEEE Software Engineering Standards, ESA (European Space Agency) software standards, CMM (Capability Maturity Model) for software, ISO-15504 (SPICE). The importance is to identify the target audience for each approach, describe the coverage of each approach, and indicate areas of overlap among them.

When a General Manager of a medium sized software company received an official letter from his long time customer informing him that he would have to prove his company had “adequate quality assurance system” if it was to stay on their supplier’s list, his first reaction was “why?”. As he went deeply into the issue he became even more concerned. He would have to comply with the complex code published by the International Organization for Standards or some other alternative software engineering standard system. This meant hiring consultants to help interpret it, writing down the company’s procedures and paying external auditors for inspection before issuing the certificate. Besides it takes a lot from the operating budgets. This concern is also shared by many others who are asking themselves: Why not simply continue to rely on a good competition to select suppliers?

The key to identifying the benefits of QA certification, lie in “reliability”. It can not be demonstrated by providing a sample of current products or even by pointing to a tradition of satisfactory supply. What if key staff leaves the company or products are redesigned? Reliability can only be demonstrated by applied systems (the policies, methods, processes and procedures) used to support the quality.

The advantages of QA certification are obvious for the customers. This allows them to make long term commitments and avoid the risks and costs of unexpected future quality problems. Also for the producing company it often means fewer warranty claims, less waste and re-work, fewer complaints, and less business “firefighting”.

What are standards in general? “Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose.” (ISO 1997). Basically if followed, they deliver an assurance of “good practice” and are not intended to be about “best practice”.

Why adopt a standard?

- As means of adopting “good practice” in your software engineering process (Standards reflect maturation process of software engineering as a formal discipline);
- As a safety net;
- As a result of the demands of your clients or procurement agencies, who may have already adopted standards;
- As a result of the adoption of other standards or SW process improvement initiatives;
- As a consequence of product/process certification requirements.

The analysis shows that if the adoption of a standard system is through a formal process (De Jure) the goals tend to take a long time to be reached, but the quality system lasts a reasonably long time. On the contrary, if it is an implicit process of agreement (De Facto), the goals can be achieved relatively rapidly, but they die quickly.

The process of standardization varies according to bodies engaged in standardization.

In this paper we try so give some practical ideas on how to achieve software process improvement and gain higher quality products through use of the existing standards and models and demonstrate how standards can be selected and tailored to meet the very different needs of the developers and the customers. We rather explore the relationship between quality and software process improvement strategies and standards. Our intention is not an explanation of the detailed structure of each of the mentioned standards/models.

## 2 Software Engineering Standards Overview

The Software Engineering Standards as normative and informative reference define how to develop software or software intensive systems. They are always document centered. They could be applied by the organization as an entity or by specific project needs.

We have classified the standards/models of our interest as follows:

International Software Engineering standards covering the life cycle processes:

**ISO/IEC/IEEE/EIA 12207**

**IEEE 1074-1995**

**ESA Standards (European Space Agency)**

**ESA PSS-05**

**ECSS-E-40(B)**

Important American standards:

**DoD Mil-Std 2915**

**IEEE SWE standards**

Quality Assurance standards

**ISO 9001(with the accompanying guidance of ISO 9000-3)**

**ECSS-M-50, ECSS-Q-80, ECSS-E-40(B)**

Software Process Assessment and Improvement standards/models:

**SEI SW-CMM (Capability Maturity Model)**

**Bootstrap**

**ISO-15504 (SPICE)**

Talking about Quality Assurance and life cycle standards, each of these approaches represents a valid philosophy of SW development. Except for ISO 9000 (ISO 9000:2000), they all cover the entire scope of SW engineering, from concept formulation to maintenance and operation. ISO 9000 starts with the system design phase and doesn’t address requirements definition phase. The IEEE SWE standards (IEEE 2000), ISO/EIA/IEEE/EIA 12207 (ISO/IEC 12207:1995), and corresponding ESA standards, and ISO 9001(revised as ISO 9001:2000) were prepared by, and for, larger and more dispersed technical communities. Any of them can be a part of a strategy for developing software “cheaper, faster and better”. Of course, they all have different purpose, scope, target audience, application domain and applicability to State of Practice. Comparing them, there could be identified some similarities, differences, areas of overlap and potential applicability (to different extent) of each approach to current software work at a SW organization. The coverage and areas of overlap are presented in Table 1.

| Subject Area | IEEE SWE | ISO/IEC 12207 and (ESA) | ISO 9001 |
|---|---|---|---|
| Program Management | X | X | X |
| SW Life Cycle Process | X | X | |
| SW Acquisition | X | X | |
| Contract Preparation | X | X | |
| Contract Review | | | X |
| System Engineering | X | X | |
| Systems Requirements | X | X | |
| Software Requirements | X | X | |
| Software Design | X | X | X |
| Implementation | X | X | X |
| System Test | X | X | X |
| Acceptance Test | | X | |
| Operations | X | X | |
| Maintenance | X | X | |
| Quality Management | | | X |
| Quality Assurance | X | X | X |
| Configuration Management | X | X | X |
| Reuse | X | | |
| Documentation | X | X | X |
| Measurement | X | | X |
| Training | | X | X |
| Software Safety | X | | |

Table 1: Coverage of the SW Engineering domain

## 3 Integrated approach in using the standards/models

No matter how clear the objectives of the customers are defined, there always exists a dilemma: “What standards/models should we apply in order to achieve our objectives?” Basically, there are two general approaches or directions. One is a general Quality Assurance approach that satisfies the expectations of the core and supporting business and the other one is concerned with the improvement strategies of the SW Process. Improving the quality of the SW Process itself is seen, more and more as a primary remedy for the non-conforming and poor quality software products. A company should consider introducing the standards/models that are dealing with the SW Process, even if that might not be a part of the certification requirements. In the Table bellow, we have tried to summarize some answers to certain dilemmas which we hope could be of help in the process of selection and decision making.

| If your need is… | For the SW Process Improvement, if your need is… |
|---|---|
| To improve your Quality management System and to improve organization performance consider revised **ISO 9000:2000**<br>To begin your SW business or quality journey, it is best to assure yourself and your customers that you are effectively managing your core business and supporting functions. To do this, consider **ISO 9001** requirements and **ISO 9004:2000** for guidelines<br>Also consider **ISO 9000** if your objective is understanding of management expectations and overall performance<br>To interpret **ISO 9000** for SW consider using any of the following SW quality system guidelines:<br>**AS 3563**<br>**ISO 9000-3**<br>**TickIt** | To define Your SW Process:<br>**ISO 12207**<br>**ECSS-E-40 (ESA)**<br>**ECSS-M-50**<br>**ECSS-Q-80**<br>**IEEE SWE set**<br>To begin your SW assessment / improvement journey or strengthen your SW process (ex. ISO 9000 registration) apply:<br>**ISO 15504 (SPICE)**<br>**SEI SW-CMM**<br>**Bootstrep**<br>Alternatively, for an integrated approach (hardware & software):<br>**SEI SW-CMMI**<br>**Trillium** |

Table 2: Two approaches in selection the standards/models

There is only a small set of internationally recognized standards. The selection process should go in several steps:

- Identify key requirements for standard;
- Negotiate requirements with the customer/procurer/contractor;
- Evaluate standards against requirements;
- Select the most appropriate standard;
- Tailor it according to your requirements;
- Monitor use and feedback.

### 3.1 The importance of ISO/IEC 12207

As we see the quality of the SW Process and its improvement as an issue of extreme importance for the essence of the quality assurance concepts, we give some highlights of this standard, published in August 1, 1995. It is voluntary; that is, it does not in itself impose any obligation upon anyone to use it. Yet, it may be imposed by an organization or SW project through internal policy directive or by contractual agreements.

The standard establishes a top-level architecture of the life cycle of software. It begins with an idea or a need and ends with the retirement of the software. The architecture is built with a set of processes and interrelationships between them. The derivation of the processes is based upon the principles of modularity and responsibility. Each process is further designed in terms of its constituent activities realized by tasks as atomic actions. The processes are divided in three groups: Primary Processes, Supporting Processes and Organizational Processes. On Figure 1 the concept of Processes that comprise this and corresponding ESA (ECSS-E-40, ECSS-M-50 and ECSS-Q-80) standards is presented.

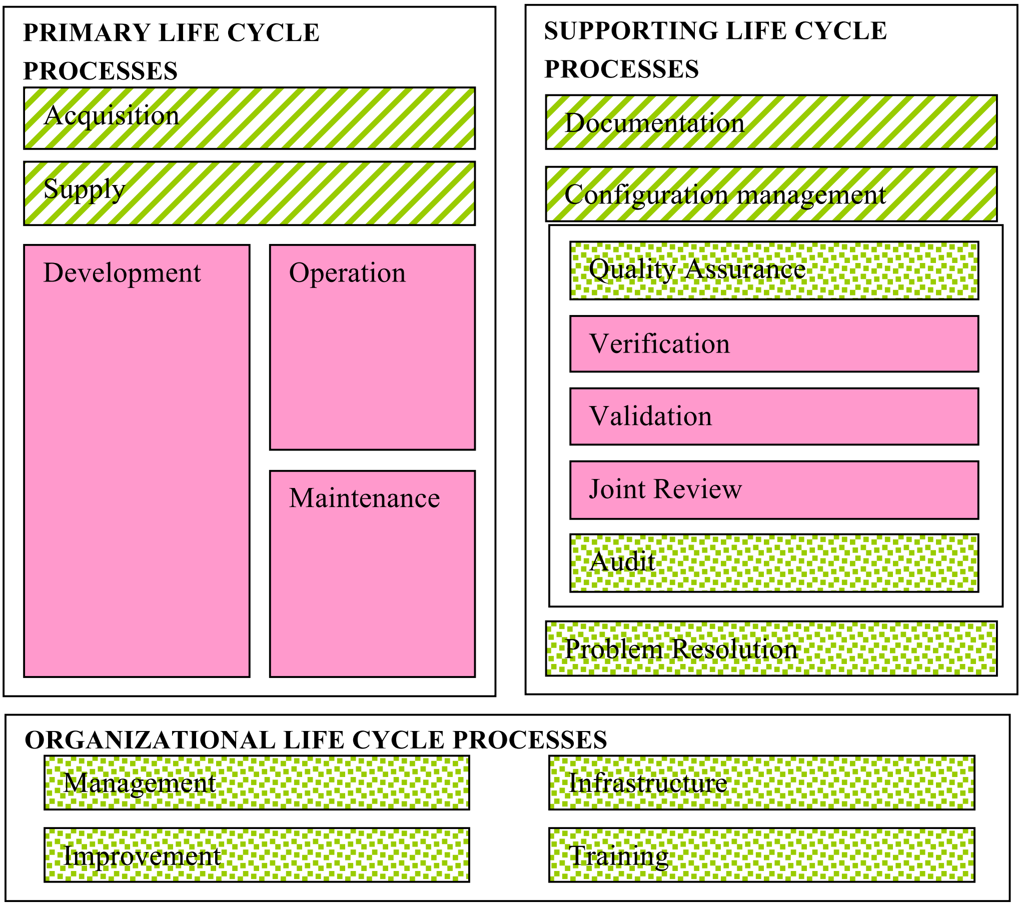

Figure 1: Life Cycle Processes in ISO/IEC 12207 and ESA standards

### 3.2 Some general considerations of the standard

**Applicability to organizations**. An organization, small or large, depending on its business purpose, can select an appropriate subset of the processes (and associated activities and tasks) to fulfill its purpose. The standard is intended to be applied by the organization internally or contractually.

**Applicability to projects.** It is written for a general, complex and large project. However it could be tailored or adapted for a smaller scale project or with lesser complexity. It is also designed to be used whether the software is stand-alone entity, or an embedded or integral part of the parent system.

**Documentation and outputs.** The standard requires certain documented outputs, but it doesn't specify the format, content or media. An organization can use its existing documentation sets or other standards with ISO/IEC 12207. This can be implemented by establishing correlation between the calls for documentation in this standard and the organization's documentation standards.

**Software metrics**. It is not a software metrics standard. It requires specifications of management indicators (such as cost expenditures) and SW attributes (reliability, maintainability, etc.) but it does not prescribe them. The standard references ISO/IEC 9126 for guidance on SW quality characteristics.

**Certification to the standard**. It does not address certification of an organization to the life cycle processes, nor does it certification criteria.

**Limitations.** The standard is not a substitute for systematic, disciplined management and engineering of software systems. It merely provides a very useful framework where processes, activities and tasks related to SW can be reasonably identified, planned and acted upon (Raghu S., 1995)

### 3.3 Relationship between Quality Assurance, Life Cycle Processes and SW Process Assessment standards

Software Assessment standards (and consequently improvement) are providing a framework for an organization to plan, manage, monitor, control and improve the acquisition, supply, development, operation and maintenance of software. Simultaneous applications of these three types of standards should be fairly non-conflicting. However, there are certain differences, which the users should be aware of before applying them on projects.

One integrated approach in using representative set of different standards (their alternative standards/models from the Table 1 or Table 2 could also be used) is shown on Figure 3.

At the top left corner, ISO 9001 represents Quality Assurance, at system level. SPICE (or SW-CMM as an alternative), (ISO/IEC 1998), represents process assessment as applied in organizations. At the bottom, ISO/IEC 12207 represents process employed throughout the life cycle of a SW product. ISO 9001 provides basis for quality assurance to the both others. ISO/IEC 12207 provides the baselines of lifecycle processes to the SPICE assessment process. It is expected that as these standards evolve in the future, their mutual consistencies would improve.

A subset (integration of ISO/IEC 12207 and SW-CMM) of this concept has been applied and tested by the author of this paper (Grceva, 2003).

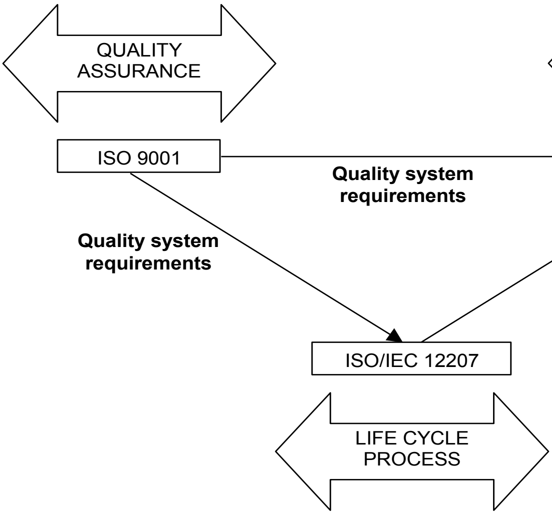

Figure 3: Relationship between Quality Assurance, Life Cycle Processes and Process Assessment standards/models

## 4 Conclusions

There are many different ways of applying the concepts of Quality Assurance, Quality Management and Principles in Software systems. The nature of the organization and the specific challenges it faces will show how and to what extent to implement them. But many organizations will find it beneficial to apply quality management principles based on internationally recognized standards and models.

Standards are about good practice, not necessarily about best practice in software development and maintenance process. If carefully targeted the adoption of standards and models can yield significant SW process improvements. Even where standards are not adopted they can be used as a benchmark. But, the organization cannot expect to adopt a standard without significant work in tailoring and customization.

In this paper, some useful classification of the subset of the existing standards and models together with selection criteria has been presented. The choice of recommended standards reflects the experience of the broader software international community, but also our own experience. The integrated use of different types of standards (Quality Assurance, Life Cycle Process and Process Assessment) directly corresponds to our previous work and positive outcomes from several pilot projects.

## 5 References

1. IEEE 1074-1995, Standard for developing Software Life Cycle Processes, IEEE Computer Society Press.
2. IEEE 2000, Software Engineering Standards Collection, IEEE Computer Society Press.

3. ISO/IEC 12207:1995, Information Technology-Software Life Cycle Processes.
4. ISO/IEC 9126: 1991, Information Technology-Software product Evaluation-Quality Characteristics and guidelines for their use.
5. ISO 9001:1994, Quality Systems-Models for quality assurance in design/development, production, installation and servicing
6. ISO/IEC 1998. Information Technology - Software Process Assessment - Concepts and introductory Guide. DTR 15504 May.
7. ISO 9000:2000, Quality Management Systems-Fundamentals and Vocabulary.
8. ISO 9001:2000, Quality Management Systems-Requirements.
9. ISO 9004:2000, Quality Management Systems-Guidelines for performance improvements.
10. Paulk MC, Curtis B, Chrissis MB, Weber CV. 1993. Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403.
11. Raghu Singh, 1995, an introduction to ISO/IEC 12207 (Tutorial.)