Quality Management of Software and Systems: Model Based Improvement Approaches
Contents

• ISO/IEC 9126
• ISO/IEC 12207: 2008 (Basis for process standards)
• EN ISO 9000: series
• CMMI (Successor of CMM)
• ISO/IEC 15504 (SPICE)
• ITIL
• CoBIT
• Personal Software Process (PSP)
• Standard for software quality focusing on software products [1].
• It defines software product quality characteristics that can be applied to:
  • Specify functional and non-functional customer and user requirements.
  • Validate the completeness of a requirements definition.
  • Identify software design and testing objectives.
  • Identify quality assurance criteria.
• These characteristics are defined within a two part quality model.

• Software Quality: describes the degree in which all defined software requirements are fulfilled. There are three different points of view to determine this: user, developer and manager.

• Quality Model: describes causal relationships between not tangible views on quality and tangible software measures:
  • Hierarchical ordered quality aspects that in the end lead to software measures.

• Two part quality model: Characteristics for Internal and External Quality:

**External and Internal Quality**

- **Functionality**
  - Suitability
  - Accuracy
  - Interoperability
  - Security
  - Functionality compliance

- **Reliability**
  - Maturity
  - Fault Tolerance
  - Recoverability
  - Reliability compliance

- **Usability**
  - Understandability
  - Learnability
  - Operability
  - Attractiveness
  - Usability compliance

- **Efficiency**
  - Time behavior
  - Resource behavior
  - Efficiency compliance

- **Maintainability**
  - Analyzability
  - Changeability
  - Stability
  - Testability
  - Maintainability compliance

- **Portability**
  - Adaptability
  - Installability
  - Co-existence
  - Replaceability
  - Portability compliance

**Internal Quality**: describes characteristics of software from an internal point of view. It specifies characteristics of interim (intermediate) products.

**External Quality**: describes characteristics of software from an external point of view. It is derived from the user’s quality requirements.
• Two part quality model: Characteristics for Quality in use:

Quality in use

- Effectiveness
- Productivity
- Safety
- Satisfaction

• **Quality in use**: depicts the user’s *point of view* with respect to *software quality* in a *specific environment* and a *context of use*. 
• Provides a common framework for **software life cycle processes** with the purpose of:
  • Providing a standardization of terms to
  • Achieve a common understanding of concepts, thus
  • Facilitating communication among stakeholders in the life cycle of a software product.

• It consists of processes, activities and tasks:
  • 44 processes distributed among 7 process groups.
  • Each process has a defined *purpose* and *outcomes*.
  • Activities and tasks that are required to achieve the expected process *outcomes* are performed.

• It covers the complete software life cycle, starting from its conception until its removal from service.

• It can be performed in combination with the ISO/IEC 15288 standard (System life cycle processes).

• **System context processes** are organized into 4 groups:
  - Agreement
  - Organizational Project-Enabling
  - Project
  - Technical
- **Software specific processes** are organized into 3 groups:
  - Implementation
  - Support
  - Reuse
• Series of standards that define fundamentals for quality management systems.

• They represent an international consensus on good quality management practices.

• These standards can be applied within all industrial sectors. In particular, the ISO/IEC 90003:2004 standard is suitable for software.

• Set of standards corresponding to the ISO 9000 series:
  • EN ISO 9000:2005 - Quality management systems - Fundamentals and vocabulary
  • EN ISO 9001:2008 - Quality management systems - Requirements
  • EN ISO 9004:2009 - Managing for the sustained success of an organization - A quality management approach
  • EN ISO 19011:2011 - Guidelines for auditing management systems
A framework for software process improvement [1]. It provides guidelines on how to select strategies toward process improvement, by:

- Establishing the current level of process maturity in an organization.
- Identifying drawbacks, which have to be improved in order to achieve a better maturity level.

The CMM defines 5 maturity levels:

1. Initial
2. Repeatable
3. Defined
4. Managed
5. Optimizing

Each level includes a set of recommended practices in several key areas.

It is believed that by performing these practices, an organization will improve its software process capability.

[1] Paulk, M.C. etal. The capability maturity model for software
• International standard dedicated to the assessment and improvement of software processes in an organization.

• It consists out of 5 parts:
  • ISO/IEC 15504-1:2004 - Concepts and vocabulary
  • ISO/IEC 15504-2:2003 - Requirements to conduct process assessments
  • ISO/IEC 15504-3:2004 - Support for performing process assessments
  • ISO/IEC 15504-4:2004 - Assistance on use of this standard for process improvement and process capability determination
  • ISO/IEC 15504-5:2006 – An exemplar process assessment model

• Analogously to ISO/IEC 12207:1995, this standard defines three process categories within nine different groups.
• Primary Life Cycle Processes:
  • ACQ: processes related to software consumers
  • SUP: processes related to software vendors
  • ENG: engineering processes used in software development
  • OPE: processes used for the operation of developed (built) software

• Organizational Life Cycle Processes
  • MAN: processes used in project management and related activities.
  • PIM: processes used to improve other processes
  • RIN: processes that supply the infrastructure required by other processes
  • REU: processes applied for the systematic reuse of software components

• Supporting life cycle processes
  • SUP: define cross-section processes for the primary life cycle processes e.g. quality assurance processes, and Validation & Verification.
• Assessments are performed by evaluating a chosen subset out of the 49 processes.

• SPICE identifies six capability levels:
  • Level 0: Incomplete
  • Level 1: Performed
  • Level 2: Managed
  • Level 3: Established
  • Level 4: Predictable
  • Level 5: Optimizing

• An assessment is conducted considering predefined Process Attributes (PA), which a process has to fulfill in each of the aforementioned levels.

• For each process, SPICE defines a Purpose, its expected Outcomes and Base Practices.

• Capability grade of a process:
  • Determined by verifying that a process outcomes fulfill the process attributes of a specific capability level.
IT Infrastructure Library (ITIL)

• Quasi-standard dedicated to manage the information technology (IT) services.
• Developed in 1989 by the Office of Government Commerce (OGC) in Great Britain.
• In particular, ITIL V3 describes a comprehensible IT-service management, which covers the planning, production and support for IT services:
  • Service Strategy
  • Service Design
  • Service Transition
  • Service Operation
  • Service Continual Service Improvement

• Unlike CMM or spice, the focus of ITIL is on production of services and not on systems development!
- **Control Objectives for Information and Related Technology (CoBIT)**, describes a widespread-controlling approach in the area of *IT-Governance*.
- *IT Governance* ensures that the IT-structures and processes of an organization are aligned with its strategies and goals (Business Alignment).
- This approach was first developed by the *Information Systems Audit and Control Association* (ISACA) and then transferred in 2000 to the *IT Governance Institute*.
- Another regulation proposed in the area of IT Governance:
  - Sarbanes-Oxley Act (SOC)

Personal Software Process (PSP)

- Motivation: the achievement of higher quality in an organization depends:
  - On the performance of the development group.
  - In turn, the performance of the group depends on the performance delivered by each one of its members.
  - Finally, the performance of each of the development group’s members depends on the practices they apply.
- The PSP is an improvement approach focused on the **performance improvement** of **single software developers**.
- PSP, organized as a self-improvement process (training program), includes a series of exercises that each developer should do (comparable to the CMM levels):
  - PSP0 – Baseline Process
  - PSP1 – Personal Planning Process
  - PSP2 – Personal Quality Management
  - PSP3 – Cyclic Personal Process
• DIN EN ISO 19011, Guidelines for auditing management systems Berlin: Beuth Verlag 2011
• ISACA, Control Objectives for Information and related technology (CoBIT®). [www.isaca.org](http://www.isaca.org)
• ISO/IEC 9126, *Software engineering – Product Quality*. Parts 1 to 4, 2001- 2004