New or Improved!

Software Engineering Standards for Quality

Annette Reilly

American Society for Quality
June 27, 2017
Annette Reilly, Ph.D., CSEP, CSDP, PMP

IEEE Computer Society Representative to ISO/IEC JTC 1/SC7, Systems and software engineering

Co-editor of ISO/IEC/IEEE 12207, Systems and software engineering—Software life cycle processes


Editor of ISO/IEC/IEEE 15289, Systems and software engineering—Content of life-cycle information items (documentation)

Editor of ISO/IEC/IEEE 24765, Systems and software engineering—Vocabulary www.computer.org/sevocab

Member, IEEE Standards Association Standards Board and IEEE Computer Society Standards Activities Board

IEEE Senior Member, IEEE Computer Society Golden Core

Lockheed Martin systems engineering manager and senior staff (retired)

annette.reilly@computer.org
“At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.”
IEEE-SA Standards Board Operation Manual (subclause 5.9.3)

ISO/IEC/IEEE 12207:2017 is in final ballot (not final)
Agenda

- Why SW Quality standards are changing
- Changes in the ISO/IEC 25000 SQuaRE series
- On the horizon... new IEEE standards
Why have standards?

- Support interoperability
- Support reliability
- Further world trade
- Promote consistent products
- Allow repeatable processes and process improvement
- Allow certification of people
- Basis for contracts and audits

Which is most useful to standardize:

Quality of software products,
Quality of software engineering processes, or
Quality of software engineers?
Standards change with changes in systems/SW engineering

- Internet, big data, software as a service, the cloud, virtual systems
- Highly complex integrated systems of systems
- Software construction through object-oriented, encapsulated, containerized methods
- Different ways of implementing software vs one-off custom/contract SW development: COTS integration, use of open source, services
- Using continuous, concurrent processes and methods (agile, DevOps, integration, test-driven development)
- Automation of software methods and tools
- Continuous and selective release methods, DevOps

“Organizations that are highly agile and responsive to market dynamics complete more of their projects successfully than their slower-moving counterparts — 75 percent versus 56 percent”

PMI 2015 Pulse of the Profession® report
Adapting Software QA for Agile?

QM/QA approach based on

- Iterative development \((not\ once\ and\ done)\)
- Frequent inspection and adaptation \((not\ big-bang\ testing)\)
- Incremental deliveries \((not\ massive\ cutovers)\)
- Requirements and solutions evolve \((not\ frozen\ plans\ and\ baselines)\)
- Through collaboration in cross-functional teams \((not\ separate\ test\ team)\)
- Through continuous stakeholder feedback \((not\ hands-off\ till\ final\ acceptance)\)

See <www.computer.org/sevocab>
What is software quality?

The quality of a system/software product is the degree to which it satisfies the stated and implied needs of its various stakeholders, and thus provides value. (ISO/IEC 25023:2016)

1. capability of a software product to satisfy stated and implied needs when used under specified conditions

2. degree to which a software product satisfies stated and implied needs when used under specified conditions
   Source: ISO/IEC 25010:2011 Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--System and software quality models

3. degree to which a software product meets established requirements
New and revised standards for Software Quality

- ISO 9001:2015 Quality Management System
- ISO/IEC 25000 series: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE): 25022, 25023
What changes in ISO/IEC/IEEE 12207 from the 2008 version

One view: Nothing in 12207 needed to change!
- Always allowed for multiple life cycle models and methods
- Never specified development/construction processes and methods

Assumptions from 1996-2008 that have changed:

- A detailed requirements spec, top-down architecture and hierarchical design decomposition are needed before development.
- Integration testing, qualification testing, acceptance testing follow development
- Software reuse and domain knowledge need to be managed and controlled at the organization level.
- A closed loop model is ideal.
- Projects have defined stages with checkpoints or gates.
- Conformity to the standard means performing all the activities

✗ “The” life-cycle model

✗ Complete, normalized input-activity-output models
  • Outcomes are not the same as Outputs

✗ Complete logical model linking various processes (waterfall or spaghetti nest?)
  • Processes can and should be done concurrently

✗ Process Assessment Model

✗ Capability-maturity model or improvement process
  • Officially, it’s all Level 1

✗ Complete listing of required work products, outputs, information items (documentation) or artifacts
  • Flexible listings (Annex B) of possible work products
The 12207:2008 process model

- **System Context process groups**
  - 25 system processes

- **Software specific process groups**
  - 18 software processes

- **4 System Context process groups**
- **3 software-specific process groups**

© Annette Reilly, 12/17/2015 | 12
Processes removed from the 2008 process model for 12207:2017

- Removed or renamed processes
- No more separate software processes

```
<table>
<thead>
<tr>
<th>System Context Processes</th>
<th>Software Specific Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Agreement Processes</strong></td>
<td><strong>SW Implementation Processes</strong></td>
</tr>
<tr>
<td>Acquisition Process (Clause 6.1.1)</td>
<td>Software Implementation Process (Clause 7.1.1)</td>
</tr>
<tr>
<td>Supply Process (Clause 6.1.2)</td>
<td>Software Requirements Analysis Process (Clause 7.1.2)</td>
</tr>
<tr>
<td><strong>Project Processes</strong></td>
<td><strong>SW Support Processes</strong></td>
</tr>
<tr>
<td>Project Planning Process (Clause 6.3.1)</td>
<td>Software Document Management Process (Clause 7.2.1)</td>
</tr>
<tr>
<td>Project Assessment and Control Process (Clause 6.3.2)</td>
<td>Software Configuration Management Process (Clause 7.2.2)</td>
</tr>
<tr>
<td>Decision Management Process (Clause 6.3.3)</td>
<td>Software Quality Assurance Process (Clause 7.2.3)</td>
</tr>
<tr>
<td>Risk Management Process (Clause 6.3.4)</td>
<td>Software Verification Process (Clause 7.2.4)</td>
</tr>
<tr>
<td>Configuration Management Process (Clause 6.3.5)</td>
<td>Software Construction Process (Clause 7.1.4)</td>
</tr>
<tr>
<td>Information Management Process (Clause 6.3.6)</td>
<td>Software Detailed Design Process (Clause 7.1.5)</td>
</tr>
<tr>
<td>Measurement Process (Clause 6.3.7)</td>
<td>Software Integration Process (Clause 7.1.6)</td>
</tr>
<tr>
<td><strong>Organizational Project-Enabling Processes</strong></td>
<td>Software Qualification Testing Process (Clause 7.1.7)</td>
</tr>
<tr>
<td>Life Cycle Model Management Process (Clause 6.2.1)</td>
<td>Software Qualification Testing Process (Clause 7.1.8)</td>
</tr>
<tr>
<td>Infrastructure Management Process (Clause 6.2.2)</td>
<td>Software Support Process (Clause 7.2.5)</td>
</tr>
<tr>
<td>Project Portfolio Management Process (Clause 6.2.3)</td>
<td>Software Validation Process (Clause 7.2.6)</td>
</tr>
<tr>
<td>Human Resource Management Process (Clause 6.2.4)</td>
<td>Software Review Process (Clause 7.2.7)</td>
</tr>
<tr>
<td>Quality Management Process (Clause 6.2.5)</td>
<td>Software Reuse Process (Clause 7.2.8)</td>
</tr>
<tr>
<td><strong>Software Reuse Processes</strong></td>
<td><strong>Software Support Processes</strong></td>
</tr>
<tr>
<td>Domain Engineering Process (Clause 7.3.1)</td>
<td>Software Support Process (Clause 7.2.5)</td>
</tr>
<tr>
<td>Reuse Asset Management Process (Clause 7.3.2)</td>
<td>Software Validation Process (Clause 7.2.6)</td>
</tr>
<tr>
<td><strong>Technical Processes</strong></td>
<td><strong>Software Reuse Processes</strong></td>
</tr>
<tr>
<td>Stakeholder Requirement Definition Process (Clause 6.4.1)</td>
<td>Domain Engineering Process (Clause 7.3.1)</td>
</tr>
<tr>
<td>System Requirements Analysis Process (Clause 6.4.2)</td>
<td>Reuse Asset Management Process (Clause 7.3.2)</td>
</tr>
<tr>
<td>System Architectural Design Process (Clause 6.4.3)</td>
<td></td>
</tr>
</tbody>
</table>
The new harmonized 12207/15288 process model

- 4 process groups, 30 processes

* New
# Split or renamed
Quality Management in 12207:2017 vs 2008

- **Identical process purpose:** to assure that products, services and implementations of the quality management process meet organizational and project quality objectives and achieve customer satisfaction

- Changes in QM process outcomes from 2008:
  - a) organization quality management policies and procedures are defined;
  - b) organization quality objectives are defined;
  - c) accountability and authority for quality management are defined; *(Life cycle model management)*
  - d) the status of customer satisfaction is monitored; **Reworded**
  - e) appropriate action is taken when quality objectives are not achieved. *(Project assessment and control)*
12207:2017 QM Outcomes

- Organizational quality management policies, objectives, and procedures are defined and implemented.
- Quality evaluation criteria and methods are established.
- New Resources and information are provided to projects to support the operation and monitoring of project quality assurance activities.
- Reworded Quality assurance evaluation results are gathered and analyzed.
- New Quality management policies and procedures are improved based upon project and organizational results.
12207:2017 QM Activities and tasks

- **Plan QM**
  - Define QM policies, objectives, procedures, responsibility and authority to implement QA, criteria and methods, provide resources and information

- **Evaluate QM**
  - Evaluate results, assess customer satisfaction, review project QA activities, monitor quality improvements

- **Perform corrective and preventive action**
  - Plan and monitor
    - *Implement corrective and preventive action under other processes*
12207:2008 QM activities and tasks not explicitly identified in 12207:2017

- Periodic reviews of project QA plans
- Take/implement corrective actions
Change in Purpose for QA

- (2008) to provide assurance that work products and processes comply with predefined provisions and plans.

2017: to help ensure the effective application of the organization’s Quality Management process to the project.

- providing confidence that quality requirements will be fulfilled.
- **Proactive analysis** of the project life cycle processes and outputs
- Assure that the product being produced will be of the desired quality and that organization and project policies and procedures are followed.
2008 QA outcomes all replaced

- a) a strategy for conducting quality assurance is developed; *(strategy is in QM)*

- b) evidence of software quality assurance is produced and maintained; *(QA records management is an activity)*

- c) problems and/or non-conformance with requirements are identified and recorded; *(management of incident and problem treatment)*

- d) adherence of products, processes and activities to the applicable standards, procedures and requirements are verified. *(Verification is another process, here, it’s QA evaluations)*
Project quality assurance procedures are defined and implemented.

Criteria and methods for quality assurance evaluations are defined.

Evaluations of the project’s products, services, and processes are performed, consistent with quality management policies, procedures, and requirements.

Results of evaluations are provided to relevant stakeholders.

Incidents are resolved.

Prioritized problems are treated.
12207:2017 Activities and Tasks

- Prepare for QA
  - QA strategy, independence of QA
- Evaluate products or services
- Evaluate processes
- Manage QA records and reports
- Treat incidents and problems
QA Strategy, (not QA Plan)

Define

- Priorities to high-impact processes and tasks
- Defined roles, responsibilities, accountabilities, authorities
- Evaluation criteria and methods, especially acceptance criteria
  - Verification, validation, monitoring, measurement, review, inspection, audit, test
- Supplier/subcontractor activities
- Problem resolution
- Process/product improvement
Agreeing on requirements
Performing peer reviews and unit/sprint testing
Analyzing problem reports and user feedback
Validating completion of corrective action
Analyzing the root cause of defects
12207:2008 detailed QA activities not in 12207:2017

- Procedures for contract reviews
- Records management procedures
- QA schedule
- SW verification, validation, review, and audit activities
- Access to QA records for the acquirer; Support and cooperation from the acquirer
- Assurance that all plans are documented, comply with the contract, are consistent, and are being executed
- All products and processes, internal SW engineering practices, development environment, test environment, and libraries comply with the contract and adhere to the plans
- Flowdowns to subcontractor
- QA of Product and process measurements
- QA of staff skills and training
Audit redefined

ISO 9000:2015 -- **Audit** systematic, independent and documented process for obtaining objective evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled.

ISO/IEC/IEEE 12207:2017 -- **Audit** independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.

**12207: 2008 -- Audit** independent assessment of software products and processes conducted by an authorized person in order to assess compliance with requirements.

- Other standards on reviews and audits
  - IEEE 1028-2008 IEEE Std on SW Reviews and Audits
  - IEEE 15288.2 IEEE Std for Technical Reviews and Audits on Defense Programs
  - ISO/IEC 20246:2017, Software and systems engineering ― Work product reviews
12207:2008 detailed software audit activities not in 12207:2017

- Detailed audit procedures
- Audits of
  - Software item to design documentation
  - Adequacy of acceptance review and testing
  - Test data, conduct of testing, resolution of discrepancies from expected results
  - User documentation conformance
  - Conformance to requirements, plans, and contract
  - Cost and schedule conformance
IEEE 730:2014 IEEE Std for SW QA Processes

- Aligned to 12207:2008 QA activities—more detailed outcomes and tasks

- “The SQA function’s responsibility is to produce and collect evidence that forms the basis for giving a justified statement of confidence that the software product conforms to its established requirements.”

- Ensure that the planned SQA activities are commensurate with product risk

- Detailed SW QA Plan (SQAP) outline and extensive checklists

- Detailed mapping to 5 process capability levels

- SQA activities in an agile environment
SQuaRE Series, 2007 to 2017

25030 Quality Requirements Division 2503n

25000, 25001 Guide, Planning & Management
25020, 21, 22, 23, 24

25040 Quality Eval Process
25041 Eval Guide
25045 Eval Module for Recoverability

25010, 25011, 25012
Quality Model Division 2501n
System/SW, Service, Data

25051 Quality of ready to use SW

Ref model, measure elements, Quality in Use, Sys & SW quality

Data Quality

Extension Division 25050 - 25099
SQuaRE Quality measurement

ISO/IEC 25010
System and software product quality

ISO/IEC 25012
Data quality

Quality characteristics

Quality sub-characteristics

ISO/IEC 25022, ISO/IEC 25023, ISO/IEC 25024

Quality measure

is defined by

Measurement function

is composed of

Quality measure elements

generates

Measurement method

is measured by

Property to quantify

includes

Target entity

ISO/IEC 25021
SQuaRE Quality Measurement

ISO/IEC 25020: Measurement reference model and guide

ISO/IEC 25020 gives guide for developing and specifying quality measures


ISO/IEC 25024: Measurement of data quality

ISO/IEC 25021 is used for developing and specifying QME for
# Quality measure elements

- Number of functions
- Number of failures
- Number of faults
- Product size
- Duration
- Number of test cases
- Number of restarts
- Number of I/O
- Number of tasks
- Number of user attempts (trials)
- Number of data items
- Number of records
- Number of requirements
- Number of user operations
- Number of system operations
- Number of languages
- Number of SW modules
- Number of interfaces
Quality in use evaluation process

Establish evaluation requirements

Purpose of evaluation
Type of product
Quality model

Contexts of use
Measures
Assessment criteria
Interpretation of measures

Specify the evaluation

Design the evaluation

Evaluate, collect data

Report

Based on ISO/IEC 25040, (SQuaRE) — Evaluation process
Newest SQuaRE Standards

ISO/IEC 25022-2016: Measurement of quality in use (Rev. of 9126-4)

ISO/IEC 25023-2016: Measurement of system and software product quality (Integration & rev. of 9126-2, 3)
ISO/IEC 25010:2011 Quality in Use Model

ISO/IEC 25010:2011, SQuaRE—System and software quality models
ISO/IEC 25010:2011 Product Quality Model

System/Software Product Quality

- Functional Suitability
  - Functional completeness
  - Functional correctness
  - Functional appropriateness

- Performance efficiency
  - Time-behaviour
  - Resource utilisation
  - Capacity

- Compatibility
  - Co-existence
  - Interoperability

- Usability
  - Appropriateness
  - Recognisability
  - Learnability
  - Operability
  - User error protection
  - User interface aesthetics
  - Acceptability

- Reliability
  - Maturity
  - Availability
  - Fault tolerance
  - Recoverability

- Security
  - Confidentiality
  - Integrity
  - Non-repudiation
  - Accountability
  - Authenticity

- Maintainability
  - Modularity
  - Reusability
  - Analysability
  - Modifiability
  - Testability

- Portability
  - Adaptability
  - Installability
  - Replaceability

ISO/IEC 25010:2011, SQuaRE—System and software quality models
ISO/IEC 25023:2016, SQuaRE—Measurement of system and software product quality
Interpretation of quality in use measures

- Conformance
- Benchmarks
- Time series (trends)
- Proficiency
- Population norms for satisfaction
**Table 6 — (User experience) pleasure measures**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Measurement function</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI-1-G</td>
<td><strong>User pleasure</strong></td>
<td>The extent to which the user obtains pleasure compared to the average for this type of system</td>
<td>( X = A )</td>
<td>Questionnaire</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A = ) Psychometric scale value from a pleasure questionnaire</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE** Examples of psychometric pleasure questionnaires are in References [14] and [23].

---

**Table 1 — Functional completeness measures**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Measurement function</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCp-1-G</td>
<td><strong>Functional coverage</strong></td>
<td>What proportion of the specified functions has been implemented?</td>
<td>( X = 1 - \frac{A}{B} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A = ) Number of functions missing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( B = ) Number of functions specified</td>
</tr>
</tbody>
</table>

**NOTE 1** Functions can be specified in a requirement specification, a design specification, a user manual or all of these.

**NOTE 2** A missing function is detected when the system or software product does not have the ability to perform a function that is specified.
ISO/IEC 25022
Economic risk mitigation measures

- Return on investment (ROI)
- Time to achieve ROI
- Business performance
- Benefits of IT investment
- Service to customers (actual vs intended level of service)
- Website visitors converted to customers
- Revenue per customer
- Errors with economic consequences
New: IEEE 2430
Nonfunctional Sizing Measurement

- Applying a nonfunctional size measurement method (based on SNAP - Software Nonfunctional Assessment Process)

- to assist in decision making for
  - better planning and estimating cost and schedule of software projects,
  - measure productivity and quality of software projects
  - analyze improvement trends

- Contact: Talmon Ben-Cnaan
  Email Address: talmonbc@amdocs.com
New: IEEE 2675, DevOps

DevOps - Standard for Building Reliable and Secure Systems Including Application Build, Package and Deployment

This standard will specify practices for groups including development, operations and other key stakeholders to collaborate and communicate effectively to build, package and deploy software and systems in a secure and reliable way.

Contact: bob.aiello@ieee.org
Get involved with standards

› Have models for your process and products

› Use standards at work
  – Get IEEE standards at
    http://www.techstreet.com/ieee

› Support a capability assessment

› Become a standards reviewer, baloter, or working group member

› Join IEEE- Standards Association and IEEE- Computer Society
Questions?