A Guide to the Forthcoming Revision of the IEEE 730 Software Quality Assurance Standard

David I. Heimann
INTRODUCTION
What is IEEE 730?

• Gives guidance and establishes requirements for Software Quality Assurance in a software project.

• The very first published software engineering standard – 1981.

• Forthcoming version of IEEE 730 greatly expands on existing version of 2002.


• SQA is one of 43 system and software process areas identified in IEEE 12207 (see next slide for chart).
## The 43 System & Software Process Areas

### System Context Processes

<table>
<thead>
<tr>
<th>Agreement Processes</th>
<th>Project Processes</th>
<th>Technical Processes</th>
<th>SW Implementation Processes</th>
<th>SW Support Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition Process (Clause 6.1.1)</td>
<td>Project Planning Process (Clause 6.3.1)</td>
<td>Stakeholder Requirements Definition Process (Clause 6.4.1)</td>
<td>Software Implementation Process (Clause 7.1.1)</td>
<td>Software Documentation Management Process (Clause 7.2.1)</td>
</tr>
<tr>
<td>Supply Process (Clause 6.1.2)</td>
<td>Project Assessment and Control Process (Clause 6.3.2)</td>
<td>System Requirements Analysis Process (Clause 6.4.2)</td>
<td>Software Requirements Analysis Process (Clause 7.1.2)</td>
<td>Software Configuration Management Process (Clause 7.2.2)</td>
</tr>
<tr>
<td></td>
<td>Decision Management Process (Clause 6.3.3)</td>
<td>System Architectural Design Process (Clause 6.4.3)</td>
<td>Software Architectural Design Process (Clause 7.1.3)</td>
<td>Software Quality Assurance Process (Clause 7.2.3)</td>
</tr>
<tr>
<td></td>
<td>Risk Management Process (Clause 6.3.4)</td>
<td>Implementation Process (Clause 6.4.4)</td>
<td>Software Detailed Design Process (Clause 7.1.4)</td>
<td>Software Verification Process (Clause 7.2.4)</td>
</tr>
<tr>
<td></td>
<td>Configuration Management Process (Clause 6.3.5)</td>
<td>System Integration Process (Clause 6.4.5)</td>
<td>Software Construction Process (Clause 7.1.5)</td>
<td>Software Validation Process (Clause 7.2.5)</td>
</tr>
<tr>
<td></td>
<td>Information Management Process (Clause 6.3.6)</td>
<td>System Qualification Testing Process (Clause 6.4.6)</td>
<td>Software Integration Process (Clause 7.1.6)</td>
<td>Software Review Process (Clause 7.2.6)</td>
</tr>
<tr>
<td></td>
<td>Measurement Process (Clause 6.3.7)</td>
<td>Software Quality Testing Process (Clause 6.4.7)</td>
<td>Software Qualification Testing Process (Clause 7.1.7)</td>
<td>Software Audit Process (Clause 7.2.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Acceptance Support Process (Clause 6.4.8)</td>
<td>Software Problem Resolution Process (Clause 7.2.8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Operation Process (Clause 6.4.9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Maintenance Process (Clause 6.4.10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software Disposal Process (Clause 6.4.11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Organizational Project-Enabling Processes

- Life Cycle Model Management Process (Clause 6.2.1)
- Infrastructure Management Process (Clause 6.2.2)
- Project Portfolio Management Process (Clause 6.2.3)
- Human Resource Management Process (Clause 6.2.4)
- Quality Management Process (Clause 6.2.5)

### Software Specific Processes

- Domain Engineering Process (Clause 7.3.1)
- Reuse Program Management Process (Clause 7.3.2)

### SQA Processes

- Software Quality Assurance Process (Clause 7.2.3)
- Software Verification Process (Clause 7.2.4)
- Software Validation Process (Clause 7.2.5)
- Software Review Process (Clause 7.2.6)
- Software Audit Process (Clause 7.2.7)
- Software Problem Resolution Process (Clause 7.2.8)
What Is Software Quality Assurance?

• A set of activities that →

1. defines and assesses the adequacy of software processes to →

2. provide evidence for a justified statement of confidence that →

3. the software processes will produce software products that →

4. conform to their established requirements.
The set of requirements that the project has:

- verified as satisfying project-specific criteria (such as clarity, suitability, and feasibility)
- validated to be faithful reflections of stakeholder requirements.

Established requirements are accepted by the project to form the basis of product development.
SQA Is Not

- Testing
- Reviewing or Auditing
- Done only at the end of development
- Reactive
- A gate or "police"
- An organizational unit (though some units may be named "SQA")
Why SQA?

- Fewer defects in the processes used to develop software.
- Fewer defects in business rules and requirements.
- Fewer defects in the software products.
- Defects are found much earlier in lifecycle and so cost far less to address.
- Reduce and eliminate waste.
- Generate confidence throughout the lifecycle that activities will go well.
You Don’t Want This

You are a lucky bug. I'm seeing that you'll be shipped with the next five releases.

http://www.amazingonly.com/cartoon/software-bugs-life/
SQA Activity Areas

I. Process Implementation
II. Product Assurance
III. Process Assurance

There are 15 SQA tasks in these 3 activity areas
Process Implementation Tasks

1. Establish the SQA Processes
2. Coordinate with related software processes
3. Plan SQA activities
4. Execute the SQA Plan
5. Manage SQA records
6. Evaluate organizational objectivity
Product & Process Assurance

- **Product Assurance**
  - ✔ Software product conforms to established requirements

- **Process Assurance**
  - ✔ Project activities conform to accurate and effective defined processes
Product Assurance Tasks

7. Evaluate plans for conformance
8. Evaluate products for conformance
9. Evaluate products for acceptability
Process Assurance Tasks

10. Evaluate lifecycle processes for conformance
11. Evaluate environments for conformance
12. Evaluate subcontractor processes for conformance
13. Evaluate product lifecycle support for conformance
14. Measure products and processes
15. Assess staff skill and knowledge
PROCESS IMPLEMENTATION
I. Process Implementation

Dilbert, by Scott Adams, via http://madhusudhan.info/Comics/Dilbert/
Define an effective SQA process that identifies **what to do** and **how to**:

1. Do it well
2. Confirm it is done right
3. Measure and track it
4. Manage and improve it
5. Encourage using it to improve quality
Enable SQA to integrate activities with other software processes, such as:

1. Verification, Validation, Review, and Audit
2. Project Planning
3. Technical Processes
4. Implementation Processes
5. Reuse Processes
6. Agreement
Task 3 – Planning the SQA Activities

• Adapt the generic SQA processes to the specific needs of the project.

• Results are documented in the Software Quality Assurance Plan (SQAP).

• This is where SQA is adapted to the specific nature of the project (e.g., Agile, CMMI, embedded, etc.)
Outline for the SQA Plan

1. Purpose and scope
2. Definitions and acronyms
3. Reference documents
4. SQA plan overview
   4.1 Organization and independence
   4.2 Software product risk
   4.3 Tools, techniques, and methods
   4.4 Standards, practices, and conventions
   4.5 Effort, resources, and schedule
5. Tasks, activities, and outcomes
   5.1 Product assurance
   5.2 Process assurance
6. Additional processes
   6.1 Contract review
   6.2 Quality measurement
   6.3 Waivers and deviations
   6.4 Task repetition
   6.5 Risks to performing SQA
   6.6 Communications strategy
7. SQA records
   7.1 Analyze, identify, collect, file, maintain and dispose
   7.2 Availability of records
Task 4 – Executing the SQA Plan

- Execute the SQAP.
- Revise the SQAP as appropriate.
- Raise non-comformances when products or processes do not conform to their requirements.
- Create and use SQA records to improve quality.
Records are created, maintained, and made available to project personnel and management.

Records aim to document that project activities:

- Are performed in accordance with project plans.
- Comply with the contract.
- Support the identification and rectification of problems, causes, and improvements.
- Enable information sharing.
Task 6 – Evaluate Organizational Objectivity

- Those who perform SQA activities must have the organizational objectivity and authority to make objective evaluations and verify problem resolutions.

- Three important aspects of objectivity are:
  - Technical Independence: Not involved in the development of the products being evaluated.
  - Managerial Independence: Not reporting to individuals responsible for product development/project management.
  - Financial Independence: Budget not controlled by individuals responsible for product development/project management.
II. Product Assurance

... and this is my grandpa. The toughest bug ever. All programmers who tried to fix him gave up and changed their careers to hairdressers.

http://www.amazingonly.com/cartoon/software-bugs-life/
1. Identify plans required by the contract.

2. Raise non-conformances when plans do not conform to the contract (or when the contractual requirements are inadequate).

3. Raise non-conformances when plans are not mutually consistent.
1. Identify products/documentation required by the contract.

2. Identify allocated requirements and ensure adequacy.

3. Ensure that evaluations of software products/documentation for conformance against the requirements are performed.
Task 9 – Evaluate Product for Acceptability

• Determine project’s understanding of conditions for product acceptance.

• Prior to delivery, evaluate the level of confidence that the software products and related documentation will be acceptable to the acquirer.

Note -- Depending on contractual agreements (e.g., Agile environments), the customers themselves may make some acceptability determinations prior to delivery.
PROCESS ASSURANCE
III. Process Assurance

Regression: "when you fix one bug, you introduce several newer bugs."

http://softwaretestingandqa.blogspot.com/ (and Calvin & Hobbes)
Task 10 – Evaluate Life Cycle Processes

- Does the **software development life cycle** conform to project plans and fit with contractual requirements?
- Does the execution of project activities conform to the project plans?
- Does the execution of project activities yield products that conform to requirements?
Task 11 – Evaluate Environments

• Do the software development environments conform to project plans?

• Do the software test environments conform to project plans?
Task 12 – Evaluate Subcontractor Processes

- Do subcontractor processes conform to requirements passed down?
- Have acquisition needs, goals, product, and service criteria been identified? Have they been met?
Task 13 – Evaluate Product Support

- Have acquirer’s expectations for product support and cooperation been established and documented?
- Have they been met?
- If the SQA process ends at delivery, how is suitable support ensured?
Task 14 – Measure Products & Processes

• Do the project measures support effective management of the software processes?

• Do the project measures meet the information needs necessary for managing effective processes?

• Do the project measures objectively determine the quality of the software products?

• Does the executed measurement process conform to the measurement plans?
Task 15 – Assess Staff Skill & Knowledge

• Do the staff, including SQA staff, assigned to the project have the knowledge, skills, and abilities to perform their assigned roles?

• Have education and training plans been developed? Are they effective?
1. Guidance for SQA Plans
2. Comparison of old & new SQA Plan outline
3. IEEE 730 and CMMI
4. IEEE 730 and IEEE 12207
5. IEEE 730 and SPICE
6. IEEE 730 and Bodies of Knowledge (BOK)
7. Industry-Specific Guidance
8. IEEE 730 and Agile
9. SQA and Very Small Enterprises (VSE)
10. Product Support
11. System vs. Software QA
12. Software Tool Validation
13. Tools, Techniques, and Methods
14. Software Integrity Levels & Assurance Cases
15. Corrective and Preventive Action
CMMI has 16 core process areas. The two that relate to quality are PPQA (Product and Process Quality Assurance) and VER (Verification).

Since CMMI does not specify a particular process flow, CMMI-conforming organizations need to design their own PPQA process.

IEEE 730 provides details for this process design.

VER process area implements product quality assurance according to the plan in PPQA. 730 covers both product and process quality assurance.

730 has associated materials with maps between 730 and CMMI.
IEEE 730 and Agile

• In Agile, the product backlog plays a role of the "contract". 730 shows how to use the product backlog in its role as a contract.

• The product SQA portion of SQA Plan specifies the Agile "done" criteria.

• Non-conformances are inserted into the backlog and addressed in the appropriate sprints.

• Evaluation of product for acceptance is a continual process in Agile, not just at end of project.

• IEEE 730 has an annex on Agile with further details.
A set of discrete values used to represent the level of risk of a software product.

The software integrity level of a product or component determines the level of rigor and quality assurance to be applied in their development and implementation.

See sample software integrity description below:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Catastrophic failure – loss of life, economic/social loss, system loss</td>
</tr>
<tr>
<td>3</td>
<td>Serious failure – loss of system, data, usability</td>
</tr>
<tr>
<td>2</td>
<td>Moderate failure – must correct and re-run</td>
</tr>
<tr>
<td>1</td>
<td>Minor failure – workaround is possible</td>
</tr>
</tbody>
</table>
Summary

- IEEE 730 provides a foundation for Software Quality Assurance, which in turns provides confidence that software products will conform to their established requirements and satisfy the customer.


- IEEE 730 can be used to prove conformance where SQA conformance is required, and to provide guidance where SQA conformance is desired.
David I. Heimann

E-mail: heimann.david@gmail.com

Phone: 617-524-4531