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 – 1979.


• IEEE 730-2014 greatly expands on the previous version of 2002; more like a whole new standard than a revision.
Why Use IEEE 730?

• Easy to use, very informative
  – Easy to follow, like a handbook
  – Gathers all the current SQA information in one place
  – Provides a clear checklist of what to do to organize the production of quality software

• Fulfills important quality purposes for an organization
  – Demonstrating conformance to the official standard for SQA
  – As a reference for developing an effective and consistent SQA process specifically pertinent to the organization
  – Obtaining information and guidance for specific questions
Who benefits when an organization adopts IEEE 730?

- **Quality managers** who are looking for guidance and streamlined implementation for SQA.
- **Project managers** who do not want poor quality to damage their schedule, budget, and ability to deliver customer-acceptable software.
- **Product managers** who want to deliver software that fully satisfies their customer’s requirements.
- **Senior managers** who want their company’s quality to be a competitive advantage, and those having customers who require a demonstration of a commitment to quality.
- **Customers and end users** who want quality software with few or no bugs.
- In short – **the entire organization!**
What is Software Quality Assurance?

- SQA is 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**.
Why Carry Out SQA?

• Results in fewer defects in the
  – Processes used to develop software.
  – Business rules and requirements.
  – Software products themselves.

• Defects are found much earlier in product lifecycle
  – Thereby costing far less in money and time address.

• Reduction and elimination of waste.

• Generate confidence throughout the project that activities are going well.
SQA Is Not

- Testing
- Reviewing or Auditing
- Reactive
- A gate or "police"
- Done only at the end of development
- An organizational unit (though some units may be named "SQA")
You Don’t Want This

Source: http://www.amazingonly.com/cartoon/software-bugs-life/. To be used for professional communications only.
SQA ACTIVITY AREAS
SQA Activity Areas

I. SQA Process Implementation

✓ Establishes the processes to carry out the two SQA activity areas below

II. Product Assurance

✓ Assures that software products conform to established requirements

III. Process Assurance

✓ Assures that project and organizational processes are accurate and effective, and that project activities conform to these processes
Process flows through a project

• In a project, stakeholder and contract requirements combine with organizational processes to create product and process requirements for the project.

• These requirements lead to software product requirements and project plans and processes, which are then used to develop the software product for delivery to the customer.

• SQA assures that the product and the process flows are smooth and sure, with dependable handoffs from one element to the next.
Process flows through a project (map)
SQA TASKS
I. TASKS WITHIN SQA PROCESS IMPLEMENTATION
BEFORE I ACCEPT THE SOFTWARE YOU WROTE UNDER CONTRACT, TELL ME WHAT DEVELOPMENT METHODOLOGY YOU USE.

WE HOLD VILLAGE MEETINGS TO BOAST OF OUR SKILLS AND CURSE THE DEVIL-SPAWNED END-USERS. SOMETIMES WE JUGGLE.

AT THE LAST MINUTE WE SLAM OUT SOME CODE AND GO ROLLER SKATING.

I WOULD FIND THIS HUMOROUS IF NOT FOR THE PIG ON MY BACK.

Source: Dilbert, by Scott Adams, via http://madhusudhan.info/Comics/Dilbert. To be used for professional communications only.
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. Use it to improve quality
Task 2 – Coordinate with Related Software Processes

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 an SQA Plan

1. Purpose and scope
2. Definitions and acronyms
3. Reference documents
4. SQA plan overview
5. Activities, outcomes, and tasks
   5.1 Product assurance
   5.2 Process assurance
6. Additional considerations
7. SQA records
Task 4 – Execute the SQA Plan

- **Execute** the SQAP.
- Revise the SQAP as appropriate.
- Raise non-conformances when products or processes do not conform to their requirements.
- Create and use SQA records to improve quality.
Task 5 – Manage SQA Records

• **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 and Assure 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. TASKS WITHIN 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 hair dressers.

Source: http://www.amazingonly.com/cartoon/software-bugs-life/
To be used for professional communications only.
Task 1 – Evaluate Plans for Conformance

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.
Task 2 – Evaluate Products for Conformance

1. Identify **products and 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 3 – 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 the development lifecycle (e.g., Agile environments), the customers themselves may be involved in prior-to-delivery acceptability determinations.
Task 4 – 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 5 – Measure Products

• Do the project measures accurately and objectively represent the quality of the software products?

• Are improvements done as a result of the product measurements effective in improving product quality?

• Do the measurements of software products satisfy the measurement requirements and conform to the measurement plans?
III. TASKS WITHIN PROCESS ASSURANCE
Regression:
"when you fix one bug, you introduce several newer bugs."

Source: http://softwaretestingandqa_blogspot.com/ (and Calvin & Hobbes)
To be used for professional communications only.
Task 1 – 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 2 – Evaluate Environments

• Do the software development and test environments conform to project plans?
Task 3 – Evaluate Subcontractor Processes

- Have project acquisition needs, goals, product, and service criteria been identified? Have they been met?
- Do subcontractor processes conform to project’s requirements and subcontractor’s own quality standards?
Task 4 – Measure 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?
- Does the executed measurement process satisfy the measurement requirements and conform to the measurement plans?
Task 5 – 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?
IEEE 730 ANNEXES
Annexes to IEEE 730

A. Mapping between IEEE 12207 and IEEE 730


C. Guidance for Creating Software Quality Assurance Plans (this is the aim of prior IEEE 730 versions and is included in this one)

D. Mapping between IEEE 730 and SPICE

E. Industry-Specific Guidance for IEEE 730

F. IEEE 730 and the Agile Development Process
Annexes to IEEE 730

G. IEEE 730 and Very Small Entities (Std 29110)

H. Software Tool Validation

I. Assessing Product Risk: Software Integrity Levels and Assurance Cases

J. Corrective and Preventive Action Processes and Root Cause Analysis Process

K. Cross-reference

L. Bibliography
IEEE 730 & Selected Software Development Lifecycles
IEEE 730 and Agile

• SQA Process Implementation activity area
  – In Agile, the product backlog plays a role of the "contract".
  – The SQA Plan is a tailored and evolving document, aiming to implement the 12 Agile principles.
  – SQA is integrated within each development team, with independence preserved through dotted-line relationships to the corporate quality management function.
  – Non-conformance issues are inserted into the product backlog and addressed in the appropriate sprints.
  – "Evaluation of product for acceptance" is a continual process in Agile, not just carried out at end of the project.
IEEE 730 and Agile

• Product Assurance activity area
  – Product assurance works through the Agile "done" criteria.
  – Non-conformance issues are inserted into the product backlog and addressed in the appropriate sprints.
  – "Evaluation of product for acceptance" is a continual process in Agile, not just carried out at end of the project.
  – Measures are of two classes
    • Development team measures focus on progress in the sprint
    • Project management measures focus on progress in the product backlog, interactions among backlog items, and overall customer satisfaction.
IEEE 730 and Agile

• Process Assurance activity area
  – SQA elements are integrated into the development teams, including test-driven development, tester and customer inclusion in development teams, continual integration, automated builds, and regression testing.
  – Strong emphasis on retrospectives after each sprint and at project completion.
  – Much stronger involvement with subcontractors, since organization is acting as an involved customer of the subcontractor.
  – Evaluation of processes, plans and environments are continuous

IEEE 730 has an annex on Agile (Annex F) containing further details.
IEEE 730 and ISO 9001

• ISO 9001 calls for answers to the following questions:
  – Tell me what you do
  – Show me where it says that
  – Will these processes help you achieve your stated objectives?
  – Prove that this is what happened

• Section 5 of IEEE 730 supplies material you can use to answer the above questions and thereby provide answers for Sections 1-8 of the ISO 9001 requirements.
IEEE 730 and CMMI

- CMMI has 16 core process areas. The ones that relate to quality are PPQA (Product and Process Quality Assurance), VAL (Validation), 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 how to design this process.
- The VAL and VER process areas implement product assurance according to the plan in PPQA.
SUMMARY
Learning objectives


2. Integration of product and process requirements through SQA.

3. The three key activity areas of SQA, and the specific tasks within each activity.

4. SQA/IEEE 730 and various SDLC’s including Agile
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.
My Software Quality Professional paper
(e-mail me for a copy)

An Introduction to the New IEEE 730 Standard on Software Quality Assurance

David L. Heimann

INTRODUCTION

Software quality is defined as how well the software meets its established requirements and stakeholders' wants, needs, and expectations. It is one of the key attributes (along with functionality, quick time to market, reasonable cost, reliability, and usability) that software must have to be a success product and a source of pride to the organization developing it. To help organizations in producing quality software, the document "IEEE Standard for Software Quality Assurance Plan" was developed in 1991 under the standard number 730 and has been updated periodically since.

The Institute of Electrical and Electronic Engineers (IEEE) has revised and updated this standard, last changed in 2002, through a technical working group that includes the author of this article. The revised standard, whose title has changed to "IEEE Standard for Software Quality Assurance Process" (IEEE 2016), and is referred to in this paper as "IEEE 730-2016," has been approved and will be released later this year (2016). However, it should be noted that the final version of the standard may include additional changes not known at the time of this publication.

With its expanded and detailed coverage, informative updates, and improved readability, the new standard goes beyond merely addressing the development of software quality assurance...
My contact information:

David I. Heimann

E-mail: heimann.david@gmail.com
Phone: 617-524-4531
www.linkedin.com/in/dheimann
www.davidiheimann.com