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

DAVID I. HEIMANN

This article provides an overview of the newly revised IEEE 730-2014 Software Quality Assurance Standard (IEEE 730). Its purpose is to guide prospective users through IEEE 730-2014 so they can skillfully use it to produce quality software. After presenting a brief description of software quality assurance (SQA) in general and IEEE 730-2014 in particular, the author describes the project components SQA monitors and how these components relate to each other. He then presents the three SQA activity areas addressed by IEEE 730-2014—SQA process implementation, product assurance, and process assurance—as well as the 16 specific activities covered in these areas. The author's aim is to provide a clear road map to IEEE 730-2014 that will enable the reader to readily grasp what SQA as described by IEEE 730-2014 is and appreciate what it does—helping them to produce better software products and services.

**Key words**

IEEE standard 730, information technology, process assurance, process implementation, product assurance, project management, quality management, quality processes, software development, software development standards, software measurements, software quality assurance, software quality assurance plans, software testing

## INTRODUCTION

Software quality is defined as how well the software meets its established requirements and stakeholder wants, needs, and expectations. It is one of the key attributes (along with functionality, quick time to market, reasonable cost, reliability, and safety) that software must have to be a successful product and a source of profit to the organization developing it. To assist organizations in producing quality software, the document "IEEE Standard for Software Quality Assurance Plans" was developed in 1981 under the standard number 730 and has been updated periodically since.

The Institute of Electrical and Electronics Engineers (IEEE) has revised and upgraded 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 Processes" (IEEE 2014) and is referred to in this paper as "IEEE 730-2014," has been approved and will be released later this year (2014). 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 annexes, and improved readability, the new standard goes beyond merely addressing the development of software quality assurance

plans (SQAP) to providing a framework for developing a complete framework of software quality assurance (SQA) processes that will implement the SQAPs. It thereby serves not only as a standard for developing and implementing SQA, but also as a guide and handbook for doing so.

This article provides an overview of IEEE 730-2014 in order to guide prospective IEEE 730-2014 users through the standard so they can better use it to produce quality software. The remainder of the first section of the article provides a brief description of SQA, an overview of IEEE 730-2014, and rationales for using the standard. The next section describes the project elements that the SQA function monitors and how they relate to each other. Subsequent sections outline the SQA activity areas and constituent activities that IEEE 730-2014 covers, and provide condensed descriptions of these activity areas. The last two sections provide a list of informative annexes contained in IEEE 730-2014, as well as summaries and conclusions.

## What Is Software Quality Assurance?

Before discussing what software quality assurance is, one should note what it is not—testing. SQA is not testing, despite a number of organizations using the terms "quality assurance," "QA," "QE," or even "SQA" to refer to their testing activities or groups. Testing is a different activity; it is addressed in IEEE 1012 – Software Verification and Validation (IEEE 2012) rather than IEEE 730-2014.

Koch (2006) states that testing and its related activities are part of quality control, which is an evaluative activity that is done after a product (or component) has been built. Therefore, he continues, quality cannot be tested into a product, it must be built into it.

So then, what in fact *is* SQA ("actual QA")? Koch states that quality assurance is proactive, consisting of activities designed to ensure that quality is built into the product. IEEE 730-2014 states that "SQA is a cascading series of activities that lead to a justified statement of confidence that a software development project yields products that fulfill the purposes that the stakeholders intend for them" (IEEE 2014). The cascade is as follows (IEEE 2014):

SQA ensures =► that the project conforms to its *processes*, and

that the *processes* consistently lead=► to *products*

that meet=► *established requirements*

that fulfill=► *stakeholder wants, needs, and expectations*

The SQA function is composed of three main process activities. One process activity, represented by the first arrow, is process assurance. Another process activity, represented by the second, third, and fourth arrows, is product assurance. The remaining process activity, SQA process implementation, is the approach by which the SQA processes are established and performed.

## What Is IEEE 730?

An applicable description of IEEE 730-2014 can be had directly from its introduction and scope clause (IEEE 2014):

> "IEEE Std 730 has been a benchmark for software quality assurance (SQA) professionals since it was first published in 1979. While previous versions of IEEE Std 730-2014 provided an SQA plan outline, this revision expands the scope of this standard to address the processes defined in software life-cycle framework standard, ISO/IEC/IEEE 12207:2008. This change in emphasis is consistent with and elaborates the process requirements in ISO/IEC/IEEE 12207:2008.
>
> "This standard establishes requirements for initiating, planning, controlling, and executing the software quality assurance (SQA) processes of a software development or maintenance project. This standard is harmonized with the software life-cycle process of ISO/IEC/IEEE 12207:2008 and the information content requirements of ISO/IEC/IEEE 15289:2011.

IEEE 730-2014 elaborates the 16 SQA tasks identified in ISO/IEC/IEEE 12207:2008 (ISO/IEC/IEEE 2008) into activities, providing specific purposes, outcomes, and tasks. Note that what is identified as a "task" in ISO/IEC/IEEE 2007:2008 is identified as an "activity" in IEEE 730-2014 (a "task" in IEEE 730-2014 is a different entity). Note also that IEEE 730-2014 splits the measurement task of 12207 into two separate activities—one for product assurance and one for process assurance. Also, IEEE 730-2014 merges the "establish the quality assurance

process" and the "additional quality management activities" tasks of 12207 into a single activity.

For each of the 16 SQA activities, IEEE 730-2014 describes (IEEE 2014):

- The *purpose* of the activity in supporting SQA
- Specific *outcomes* that fulfill the purpose
- Specific *tasks* whose completion will achieve the outcomes

For example, in the "Manage SQA Records" activity (see clause 5.3.5 in IEEE 730-2014 [IEEE 2014]), the *purpose* is:

> "Create records of SQA activities, outcomes, and tasks; manage and control these records; make these records available to project stakeholders."

The *outcomes* are:

- "Records related to SQA activities, outcomes, and tasks are created.
- "Records are maintained and stored in accordance with appropriate organizational, regulatory, and project plan requirements.
- "Records are made available to project stakeholders as specified by the contract and the SQAP."

The *tasks* are:

1. "Create records as required by the SQAP. These records capture findings of SQA activities and tasks and provide evidence that SQA activities and tasks were performed.
2. "Maintain records according to trustworthiness, and security and privacy requirements.
3. "Identify records of quality assurance activities as accessible, deliverable, or internal use only to avoid contract noncompliance or inadvertent transfer of intellectual property.
4. "Maintain the integrity of the SQA function's records through a document control system to prevent their modification or inadvertent removal and release.
5. "Supply specific records to authorized stakeholders defined in the contract. Records are made available subject to confidentiality and other constraints. The contract specifies what the acquirer will receive from the supplier; the SQAP specifies which SQA function records the acquirer's internal organizations will receive."

IEEE 730-2014 also provides background information about key concepts of SQA, such as:

- The interplay between project and organizational responsibility
- The relationship between SQA and requirements
- An overview of SQA conformance relationships
- Acquirer and supplier perspectives
- Defining the SQA role
- Software product risk
- Software process improvement

In addition, IEEE 730-2014 includes many informative annexes. A detailed list is provided later in this article.

## Why Use IEEE 730?

There are three main reasons an organization should use IEEE 730-2014:

1. *Demonstrate conformance to the official standard for SQA.* For projects where the customer, supplier organization, and/or regulatory bodies require such conformance, the value of conformance is clear. IEEE 730-2014 has informative annexes that discuss its use in the medical device and nuclear industries. Another area where conformance is often required is the development of critical software, where failure can impact the safety of people or the environment, or cause substantial financial losses. IEEE 730-2014 can be used to define a level of conformance required for critical software in order to establish a stakeholder-required level of reliability, safety, and security (IEEE 730-2014 has an informative annex, Annex I, going into these issues in greater detail).
2. *Use IEEE 730 as a reference for developing an effective and consistent SQA process tailored to the specific needs of an organization.* Many organizations know that they need processes to follow in order to maximize the likelihood that their software products will satisfy their customers, especially in an environment of constantly changing market conditions. They know they need to be constantly analyzing and evaluating the necessarily incomplete view that these processes present and adjusting the processes' inputs, algorithms, models, approaches, and outputs on the basis of organizational experience

and learning. To such organizations, IEEE 730-2014 provides a minimum set of process areas to meet the standard, providing for each area outcomes to aim for and tasks to carry out. Conformance to IEEE 730-2014 allows an organization to develop an SQA structure in which it can function with reasonable confidence for a quality product (for those organizations using an agile development to address these changing market conditions, IEEE 730-2014 provides an annex, Annex F, showing how to integrate SQA into an agile environment).

3. *Obtain information and guidance on specific questions.* IEEE 730-2014 also serves as a powerful and easy-to-follow guide to producing a quality software product. Each of the 16 activity areas described in IEEE 730-2014 (in clause 5 of the standard) shows specific outcomes to aim for and tasks to follow to obtain these outcomes, while the clause on key concepts of SQA (clause 4 of the standard) and the standard's annexes provide important background material.

## PROJECT COMPONENTS AND THEIR INTERRELATIONSHIPS WITHIN SQA

SQA oversees a set of components consisting of requirements, plans, processes, products, and others. The components, identified in clauses 4.3 and 4.4 of IEEE 730-2014, are shown in Table 1 and diagrammed in Figure 1. A primary goal of SQA is to ensure that each of these components conforms to the component(s) immediately upstream from it (see Table 2).

## THE ACTIVITIES AND PROCESS AREAS OF SQA

SQA carries out 16 activities on the components listed in Table 1, each of which is described in detail in clause 5 of IEEE 730-2014. These activities are grouped into three process areas, shown in Table 3 (derived from IEEE 730-2014 [IEEE 2014]), described briefly in later sections of this article, and addressed in full detail in IEEE 730-2014 itself. IEEE 730-2014 also contains informative annexes providing guidance on many related topics.

**TABLE 1** Project components

| |
|---|
| 1. Stakeholder wants, needs, and expectations |
| 2. Rules, regulations, and laws |
| 3. Contract requirements *(requirements specified in the contract)* and stakeholder requirements *(expressed or implied requirements from the stakeholder)*, together forming *established requirements* |
| 4. Organizational processes (*organizational-level activities applied to the project*) |
| 5. Process requirements *(the processes the project will use to produce the project outcomes)*, which lead to:<br>a. Required plans<br>b. Project processes<br>c. Other processes (environments, subcontractors, skills, and knowledge) |
| 6. Product requirements *(the functions the product is mandated to perform and attributes the product is mandated to possess)*, which lead to:<br>a. Software requirements *(product requirements that relate to software)* |
| 7. Execution of activities |
| 8. Products:<br>a. Software<br>b. Documentation<br>c. Customer support |



**TABLE 2** A goal of SQA is to ensure each of these components conforms to the component immediately upstream

| This component → | Conforms to this component |
|---|---|
| Contract and stakeholder requirements | Stakeholder wants, needs, and expectations |
| Contract and stakeholder requirements | Rules, regulations, and laws |
| Process requirements | Contract and stakeholder requirements |
| Process requirements | Organizational processes |
| Project processes | Process requirements |
| Required plans | Process requirements |
| Environments, subcontracts, and skills | Process requirements |
| Product requirements | Contract and stakeholder requirements |
| Software requirements | Product requirements |
| Required plans | Software requirements |
| Execution of activities | Required plans |
| Execution of activities | Project processes |
| Execution of activities | Environments, subcontracts, and skills |
| Software, documents, support | Software requirements |

**TABLE 3** SQA activity areas and activities

| SQA Activity | Section in IEEE 730 |
|---|---|
| **SQA Process Implementation** | **5.3** |
| Establish the SQA process | 5.3.1 |
| Coordinate with related processes | 5.3.2 |
| Document SQA planning | 5.3.3 |
| Execute the SQA plan | 5.3.4 |
| Manage SQA records | 5.3.5 |
| Evaluate organizational independence and objectivity | 5.3.6 |
| **Product Assurance** | **5.4 (and 5.4.1)** |
| Evaluate plans for conformance | 5.4.2 |
| Evaluate product for conformance | 5.4.3 |
| Evaluate product for acceptability | 5.4.4 |
| Evaluate product life-cycle support for conformance | 5.4.5 |
| Measure products | 5.4.6 |
| **Process Assurance** | **5.5 (and 5.5.1)** |
| Evaluate life-cycle processes and plans for conformance | 5.5.2 |
| Evaluate environments for conformance | 5.5.3 |
| Evaluate subcontractor processes for conformance | 5.5.4 |
| Measure processes | 5.5.5 |
| Assess staff skill and knowledge | 5.5.6 |



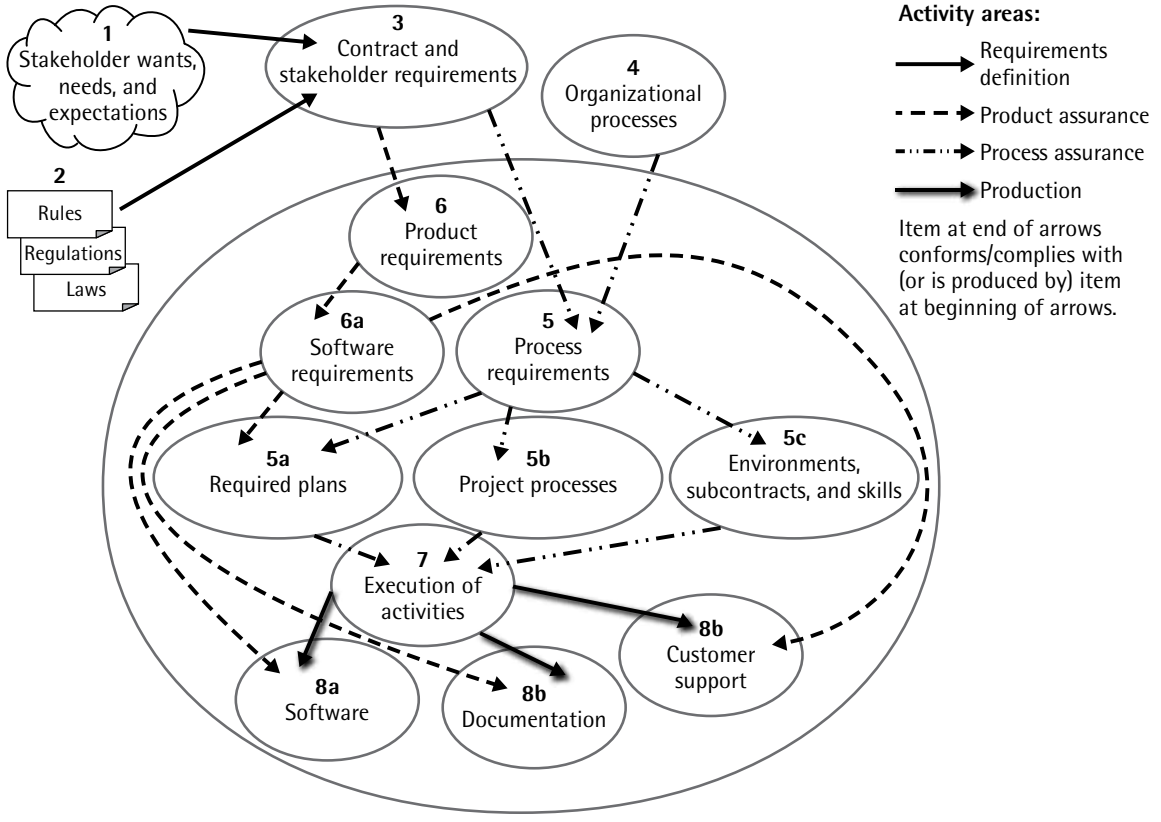
# SQA PROCESS IMPLEMENTATION (clause 5.3 in IEEE 730-2014)

To carry out SQA, a project establishes SQA processes and functions (which may be based on existing organizational processes and functions), documents them in an SQA plan, and then executes the plan.

Establishing and implementing the SQA process enables key evidence of software quality assurance to be produced, maintained, and acted upon. It also establishes the SQA function's organizational independence, allowing it to act as a voice for software quality.

Specifically, the project carries out the following:

**FIGURE 1** Items and conformance relationships within the SQA process



1. Establish the SQA function for the project (clause 5.3.1):
   a. Determine the process requirements and consequent processes SQA needs to track.
   b. Determine the product requirements that SQA needs to track.
2. Coordinate with related project and organizational processes (clause 5.3.2).
3. Create an SQA plan (clause 5.3.3).
4. Carry out the SQA plan (clause 5.3.4).
5. Create and use a record-keeping system to document SQA activities (clause 5.3.5).
6. Ensure SQA's organizational independence clause (clause 5.3.6).

Note that carrying out a retrospective at the end of the project will benefit succeeding projects and the overall organization in general.

## Establish the SQA Processes (clause 5.3.1 in IEEE 730-2014)

Establishing the SQA processes is key to carrying out SQA and thereby producing quality software. The SQA processes are established by carrying out these steps:

- Establish the SQA policy and its role within the project and larger organization.
- Identify the components—requirements, processes, plans, tasks, and products—that SQA will evaluate.
- Assign people and other resources to SQA.
- Establish a method for overseeing SQA and providing feedback and lessons learned.
- Establish SQA's organizational independence—technical, managerial, and financial.

If the organization has established SQA functions and processes, the project can use these to establish the SQA function and processes within the project. If not, the project can be guided by clause 4 in IEEE 730-2014—"Key Concepts of Software Quality Assurance," which provides an overview of SQA issues, as well as by what previous projects in the organization have done with respect to SQA. Clause 4 of IEEE 730-2014 includes topics such as:

- Organizational considerations of the SQA function
- Relationship of software quality to stakeholder and other requirements
- Elements of SQA activities (see Table 3)
- Software product risks
- Relationship between software and systems software process improvement

## Coordinate With Related Software Processes (clause 5.3.2 in IEEE 730-2014)

ISO/IEC/IEEE 2007:2008 has many (39) other processes related to the software life cycle (ISO/IEC/IEEE 2008), and a project will use many of these processes. Because the SQA function is responsible for overall process coordination, it needs to work with project and organizational management to determine which of these processes apply to the project and how to coordinate SQA activities with them. These processes are outlined in the Appendix.

Of these related software processes, the ones most closely related to the SQA process itself are four of the software support processes:

- Software verification
- Software validation
- Software review
- Software audit

These four closely related software processes are specifically mentioned in clause 5.3.2 of IEEE 730 as part of "Software Support Processes," and are addressed in detail in IEEE 1012 (IEEE 2012) and IEEE 1028 (IEEE 2008).

Other related software processes mentioned in clause 5.3.2 of IEEE 730-2014 are:

- Software implementation process
- Software reuse processes
- Agreement processes
- Project processes
- Technical processes
- Quality management process
- Life-cycle model management process

A variety of software maturity models have developed key processes that demonstrate capability at various maturity levels. These models include SQA as one of their key processes. They fulfill the "coordinate with

related software process areas" activity of IEEE 730-2014 by specifying SQA's relationship to the other key processes the models take into account.

An example is the following process suite. While it is that of CMMI Levels 2 and 3 (SEI 2010), other models such as Software Process Improvement and Capability Determination (SPICE) (ISO 2003) and ISO 9000-3/ISO 9001 (ISO 2004/ISO 2008) contain similar provisions.

Processes at the project/repeatable maturity level (see Level 2 of SEI 2010 for details):

- Configuration management
- Measurement and analysis
- Product and process quality assurance (this is CMMI's terminology for the SQA process)
- Project planning, monitoring, and control
- Requirements management
- Supplier agreement management

Processes at the organizational/managed maturity level (see Level 3 of SEI 2010 for details):

- Decision analysis and resolution
- Integrated project management
- Organizational process definition
- Organizational process focus
- Organizational training
- Product integration
- Requirements development
- Risk management
- Technical solution
- Validation
- Verification

## Document SQA Planning (clause 5.3.3 in IEEE 730-2014)

The central part of SQA is the SQAP, which documents the plans for SQA activities on the project. The SQAP identifies the SQA activities and tasks as well as any tailoring of organizational quality processes to the specific project, and includes specific needs of the project as well as the overall quality approach.

Clause 5.3.3 of IEEE 730-2014 prescribes the outline to follow in writing the SQAP, and Annex C provides detailed guidance for developing the SQAP according to that outline. The outline is shown in Table 4.

The previous version of IEEE 730, written in 2002, focused specifically on the writing of an SQAP. That information is now covered in clause 5.3.3 (which includes the outline in Table 4) and Annex C of the current IEEE 730-2014 standard (IEEE 2014), with the rest of the current standard and its annexes being new SQA material not contained in the previous version. For those familiar with the previous version of IEEE 730, Annex B in the current standard provides a map between the SQAP outline of the previous standard and that of the current one.

**TABLE 4** Outline to follow when writing the SQAP

| |
|---|
| **1. Purpose and scope** |
| **2. Definitions and acronyms** |
| **3. Reference documents** |
| **4. SQA plan overview:** |
| 4.1 Organization and independence<br>4.2 Software product risk<br>4.3 Tools<br>4.4 Standards, practices, and conventions<br>4.5 Effort, resources, and schedule |
| **5. Activities, outcomes, and tasks:** |
| 5.1 Product assurance:<br>5.1.1 Evaluate plans for conformance<br>5.1.2 Evaluate product for conformance<br>5.1.3 Evaluate product for acceptability<br>5.1.4 Evaluate product life cycle support for conformance<br>5.1.5 Measure products<br>5.2 Process assurance:<br>5.2.1 Evaluate life cycle support for conformance<br>5.2.2 Evaluate environments for conformance<br>5.2.3 Evaluate subcontractor processes for conformance<br>5.2.4 Measure processes<br>5.2.5 Assess staff skills and knowledge |
| **6. Additional considerations:** |
| 6.1 Contract review (including verifying and validating the contract itself and the consequent established requirements)<br>6.2 Quality measurement<br>6.3 Waiver and deviations<br>6.4 Task repetition<br>6.5 Risks to performing SQA<br>6.6 Communications strategy<br>6.7 Nonconformance process |
| **7. SQA records:** |
| 7.1 Analyze, identify, collect, file, maintain, and dispose<br>7.2 Availability of records |

## Execute the SQA Plan (clause 5.3.4 in IEEE 730-2014)

After the SQA plan that documents the SQA planning effort has been written, it needs to of course be carried out. Executing the plan includes:

- Carrying out the steps shown in the SQA plan
- Revising the SQA plan as needed to reflect changes in the project
- Generating reports and records to document the efforts
- Raising and resolving nonconformances that arise when actual outcomes do not agree with contract requirements or expectations

The last bullet item, raising and resolving nonconformances, is a key part of the SQA function, since this is the main mode by which SQA gets its work completed.

## Manage SQA Records (clause 5.3.5 in IEEE 730-2014)

For SQA to succeed, a project or organization must establish and manage records that show exactly what has been done, the results that were obtained, and the follow-up carried out to rectify problems and nonconformances. This implementation includes:

- Creating the records
- Updating them as activities occur
- Maintaining them and preserving their integrity
- Making them readily available to project staff, management, acquirer representatives, and subcontractor staff if subcontracting is involved

## Evaluate Organizational Objectivity and Independence (clause 5.3.6 in IEEE 730-2014)

To carry out SQA's oversight and assurance role, the people responsible for performing the SQA function need to have a role within the organization that provides an unimpeded communication mechanism with organizational management. Also, they need to have the resources and authority to make objective evaluations, and initiate, effect, and verify problem resolutions.

This objectivity and independence takes three forms (IEEE 2012):

- Technical independence – SQA is carried out by people who are not involved in the development of the system or its elements. Such people are more likely than those heavily involved in development to detect subtle defects or nonconformances.
- Managerial independence – Those responsible for the SQA effort are vested in an organization separate from the development and program management organizations. This allows SQA to submit its results to program management without restrictions (such as prior approval) or adverse pressures from the development group or project management.
- Financial independence – The SQA budget is vested in an organization independent of the development organization. This protects the SQA efforts from diversion of its funding or similar financial pressures.

# PRODUCT ASSURANCE (clause 5.4 in IEEE 730-2014)

Product assurance establishes confidence in the software products (which also includes documentation, services, and other related outputs). This is done by compiling evidence for a justified statement of confidence that the products fulfill their established requirements as shown in the contract and related documents.

Product assurance contains these core activities:

- Ensuring that the project plans are documented, comply with the established requirements (which in turn comply with the contract), are mutually consistent, and are being executed as required, especially against the established requirements (clause 5.4.2)
- Ensuring that the products, which include the software itself, user documentation, and customer support, comply with the contract and adhere to the plans (clause 5.4.3)
- Ensuring that the products comply with contractual requirements and are acceptable to the customer (clause 5.4.4)
- Ensuring that the acquirer and other parties are provided with the proper level of product life-cycle support (clause 5.4.5)
- Ensuring that the product measurements are in conformance with established policies and procedures (clause 5.4.6)

## Evaluate Plans for Conformance (clause 5.4.2 in IEEE 730-2014)

Much as blueprints and architectural diagrams are identifiable products of an effort to build a house (and, like other products, can be purchased separately from the homebuilding effort), plans are a product of the software development effort. Correct and consistent plans that are compatible with the requirements are necessary parts of successfully carrying out a quality project.

Assurance of plans includes:

- Identifying the required plans
- Evaluating that they are consistent with each other
- Evaluating that they fulfill the contract and the established requirements
- Raising nonconformances when the aforementioned does not happen

## Evaluate Software Products for Conformance (clauses 5.4.3 and 5.4.5 in IEEE 730-2014)

Assurance of the software products, that is, evaluation for conformance to established requirements, (as well as customer acceptability) is the key part of product assurance and an important part of SQA in general. It ensures that the products that result from the development effort fit the specifications and designs (that is, things are done right), and fulfill the established and correct requirements (that is, the right things are done).

Much of the relationship of SQA to other software development processes, especially verification, validation, review, and audit, are focused on monitoring and raising nonconformance issues in order to ensure that these conformances happen.

To the extent that testing has a connection to QA/SQA, product assurance is the area to which it connects.

Assurance of software products consists of:

- Identifying the required software products, related documentation, and customer support items
- Allocating the established requirements to the specific products, documentation, and support items
- Evaluating (verifying, validating, or reviewing) that each product, documentation, and support item fulfills the requirements allocated to them
- Raising nonconformances when products, documentation, or support items do not fulfill the requirements allocated to them

## Assurance of the Product's Acceptability to the Customer (clause 5.4.4 in IEEE 730-2014)

A product can fulfill each individual requirement and be free of defects, but nonetheless not provide what the customer (reasonably) wants and is therefore not acceptable to them. This can be because the requirements were not clear, the customer's business situation has changed, the original situation has been "overcome by events," or a number of other reasons. Therefore, a mechanism is needed to ensure (as much as possible) that prior to delivery the product is acceptable to the customer, so that action can be taken in a timely manner if this is not the case.

Acceptance assurance steps include:

- Documenting the supplier's understanding of the criteria for product acceptance
- Determining whether the customer has the means to determine the criteria for product acceptability
- Documenting the customer's criteria for product acceptance
- Determining whether the product conforms to the acceptance criteria
- Raising nonconformances when the criteria are not conformed to or when the products do not satisfy contractual obligations
- Optionally developing by the supplier and customer a mechanism to establish acceptability prior to product delivery

In agile or other incremental processes, multiple deliveries of product may be made to the customer. In that case, acceptability assurance activities will happen more than once.

## Product Measurements (clause 5.4.6 in IEEE 730-2014)

To provide the evidence that the product assurance requirements are met (as well as process assurance requirements), organizations need to develop a measurement system to generate the appropriate evidence. Such measurements:

- Are carried out on the software products
- Accurately represent product quality
- Conform to the project's processes, plans, standards, and procedures, with nonconformances being raised when the measurement activities do not do so
- Produce reports that share measurement information with management and stakeholders
- Fully address the SQA function's measurement needs

Creating such measures is a result of a measurement process that includes:

- Identifying the measurement procedures established by the project (or organization)
- Determining whether these measurements are representative of product quality attributes, and consistent with project standards and procedures
- Analyzing whether the measurement procedures are sufficient to satisfy the project's product measurement requirements
- Having done the previous steps:
  - Analyzing the product measurements to find the gaps between measurements and expectations and recommend improvement to close these gaps
  - Evaluating measurement results to determine whether improvement steps have been effective

The steps described here pertain as well to products developed by subcontractors.

# PROCESS ASSURANCE (clause 5.5 in IEEE 730-2014)

In process assurance, organizations make sure that the processes used for the software conform to the project's process requirements (which in turn conform to contract/stakeholder requirements and organizational processes).

Process assurance contains these core activities:

- Ensuring that the project life-cycle processes and the project plans conform to the project's process requirements, which in turn conform to the contract (clause 5.5.2)
- Ensuring that the project environments conform to the project processes and adhere to the plans (clause 5.5.3)
- Ensuring that subcontracts and subcontractors follow the process requirements and deliver conforming products (note that the relationship of the supplier to the subcontractor is similar to that between the main supplier and the acquirer) (clause 5.5.4)
- Ensuring that the process measurements are in conformance with established policies and procedures (clause 5.5.5)
- Ensuring that the project staff has the necessary skills, knowledge, and training (clause 5.5.6)

## Evaluation of the Life-Cycle Processes and Plans for Conformance to Requirements (clause 5.5.2 in IEEE 730-2014)

Making sure the project processes and plans meet the process requirements is the central part of process assurance. This is similar to making sure products meet product requirements on the product assurance side.

For the most part, process conformance is carried out through periodic reviews and audits, though failures of project activities to follow processes may themselves cause problems that will eventually lead to specific root-cause analyses and other problem-solving techniques.

Either the acquirer or organization management may generate specific process requirements that the project processes must conform to. This could arise, for example, from various levels of risk or various levels of regulatory compliance that cause a demand for certain processes to be carried out, or from an overall organizational policy dictating specific given processes.

For stability in the face of changing requirements, especially in agile or other iterative and incremental life cycles, part of this conformance activity involves setting specific project process change management procedures.

## Evaluation of Software Environments (clause 5.5.3 in IEEE 730-2014)

Software engineering practices, software engineering and test environments, and libraries are necessary for a project to develop quality products.

They may be explicitly specified in the contract or project plans. To the extent that they are, IEEE 730 specifies that the SQA function reviews them to ensure that they are in conformance with the contract or plans, and raises nonconformances if they are not.

Even if they are not explicitly specified, IEEE 730 calls for the software engineering practices, environments, and libraries to be reviewed nonetheless to ensure that they are in conformance with overall project processes, with nonconformances being raised if they are not.

## Evaluation of Subcontractor Activities for Conformance (clause 5.5.4 in IEEE 730-2014)

In many projects, some of the product development activities are subcontracted to organizations and companies outside the supplier's own organization. Even though the supplier is contractually responsible to the acquirer for the resulting products, the supplier has no direct control over the subcontractor. The supplier therefore needs to pass along the same product and process requirements to the subcontractor that they themselves have to satisfy, and assure themselves that the subcontractor has conformed to them.

In this activity, the SQA and the SQAP identify product and process requirements that are allocated to the subcontractor from the supplier. SQA then determines whether the subcontractor has established their product and process requirements, whether these conform to the supplier's product and process requirements, and whether the subcontractor is indeed conforming to their established requirements.

## Evaluation of Process Measurements for Conformance (clause 5.5.5 in IEEE 730-2014)

As with product assurance, to provide the evidence that the preceding process assurance requirements are met, organizations need to develop a process measurement system that obtains data to generate the appropriate evidence. In a similar manner as with the product measurements, such measurements:

- Are carried out on the software processes
- Accurately represent process quality
- Conform to the project's processes, plans, standards, and procedures, with nonconformances being raised when the measurement activities do not do so
- Produce reports that share measurement information with management and stakeholders
- Fully address the SQA function's measurement needs

In a similar manner, creating and using such measures is a result of a measurement process that includes:

- Identifying the measurement procedures established by the project (or organization)
- Determining whether these measurements are representative of process quality attributes, and consistent with project standards and procedures
- Analyzing whether the measurement procedures are sufficient to satisfy the project's process measurement requirements
- Having done the previous steps:
  - Analyzing the process measurements to find the gaps between measurements and expectations and recommend improvement to close these gaps
  - Evaluating measurement results to determine whether improvement steps have been effective

Note that the steps described here pertain as well to processes used by subcontractors.

## Evaluation of Staff Skills and Knowledge for Conformance (clause 5.5.6 in IEEE 730-2014)

Much as quality tools are necessary for a quality product, so are quality skills and knowledge on the part of the management and staff of the project and organization. IEEE 730-2014 prescribes that the organization identify the needed skills and knowledge, evaluate whether and to what extent the management and staff have them, and to the extent they do not, to put plans into effect to make sure management and staff acquire these necessary skills and knowledge.

IEEE 730-2014 also prescribes that the organization continually carry out development plans to ensure staff possession of all required skills and knowledge.

To further these outcomes, the SQA function does the following:

- Audits the organization staff's skills and knowledge, compares it to those required by the project, and identifies any gaps.
- Determines whether development programs are in place to address these gaps and whether they are doing the job.
- Determines any changes in required skills and knowledge, as well as those needed by new members coming on board. They also monitor personnel training records on a regular basis.

## Defect Detection

In software there is no way of counting how many defects actually escaped a defect detection technique. The types of defects detected by subsequent defect detection techniques can be examined to approximate the number of actual escapes. For example, the end of the first defect detection technique (requirement defect detection—typically a requirement peer review) there are no known escapes.

# INFORMATIVE ANNEXES IN IEEE 730-2014

In addition to the main text of the standard, IEEE 730-2014 contains informative annexes on many related topics. These annexes include (IEEE 2014):

- Annex A: Mapping Between IEEE/ISO/IEC 12207-2011 clause 7.2.3 Outcomes and IEEE 31 Standard 730:2014.
- Annex B: Mapping Between SQA Plan Outlines Contained in IEEE 730-2002 and IEEE 730-2014
- Annex C: Guidance for Creating Software Quality Assurance Plans
- Annex D: Mapping IEEE 730-2014 and ISO/IEC 15504 (SPICE)
- Annex E: Applying IEEE 730-2014 – Industry-Specific Guidance
- Annex F: SQA Activities and Their Relationship to the Agile Development Process
- Annex G: Mapping Between IEEE 730 and ISO/IEC 29110 Standard for Very Small Entities
- Annex H: Software Tool Validation
- Annex I: Assessing Product Risk: Software Integrity Levels and Assurance Cases
- Annex J: Example Corrective and Preventive Action Process and Root Cause Analysis Process
- Annex K: Cross-Reference
- Annex L: Bibliography

# SUMMARY AND CONCLUSIONS

SQA is a key part of the software development process. It is a cascading series of activities and processes whose aim is to ensure that the software will pass its tests; conform to its established requirements; meet stakeholder wants, needs, and expectations; and satisfy the customer.

IEEE 730-2014 goes beyond its former scope of providing a standard for SQAPs. It expands this scope by providing detailed normative requirements in all three process areas of SQA—process implementation, product assurance, and process assurance.

IEEE 730-2014 serves these key functions:

- Providing a rubric to show conformance to SQA standards where such conformance is required by the customer or external regulators
- Providing a rubric to show conformance to SQA standards where such conformance strongly attracts business from current and potential customers
- Providing detailed information on key SQA issues
- Providing an easy-to-follow road map to developing and implementing an effective SQA process

This article provides summaries of each component of the standard's principal clause—clause 5; Software Quality Assurance Process, as well as describing the major project components addressed in SQA along with their mutual interrelationships. This article is intended for use as an introduction and handbook to the overall standard, which in turn can be used as a handbook for the overall SQA process itself, leading to better quality software.

## ACKNOWLEDGEMENTS

The author would like to thank Kate Pulnik for editing the original manuscript and Sue Carroll, Byron Frank, Eva Freund, and Steve Rakitin for their comments.

## REFERENCES

IEEE. 2008. IEEE 1028-2008 – *IEEE Standard for Software Reviews and Audits*. New York: Institute of Electrical and Electronic Engineers (IEEE).

IEEE. 2012. IEEE 1012-2012 – *IEEE Standard for System and Software Verification and Validation*. New York: Institute of Electrical and Electronic Engineers (IEEE).

IEEE. 2014. *IEEE Standard 730 for Software Quality Assurance Processes*. New York: Institute of Electrical and Electronic Engineers (IEEE).

ISO. 2003. *ISO/IEC 15504-2:2003 Information technology – Process assessment – Part 2: Performing an assessment*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO. 2004. *ISO 90003:2004 Software engineering – Guidelines for the application of ISO 9001:2000 to computer software*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO. 2008. *ISO 9001:2008 Quality Management Systems: Requirements*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC/IEEE. 2008. *ISO/IEC/IEEE 12207-2008 Systems and Software Engineering – Software Life Cycle Processes*. Institute of Electrical and Electronic Engineers (IEEE).

Koch, Alan S. 2006. Connecting with quality – Testing is not quality assurance. Project Connections. Available at: http://www.projectconnections.com/articles/080306-koch.html.

SEI. 2010. *CMMI for development, version 1.3*. Pittsburgh, PA: Software Engineering Institute (SEI), Carnegie Mellon University.



## BIOGRAPHY

**David Heimann** received a bachelor's degree in mathematics from City College of New York, a master's degree in mathematics from Purdue University, and a doctorate in computer science from Purdue. He has held positions in government, industry, and academia, and has worked in the fields of software analysis and metrics, software process improvement, database management systems, reliability modeling, simulation, and probabilistic modeling.

He has published numerous articles, including the relationship between software analysis and complexity, system reliability and availability modeling, case studies on software metrics implementation, and using metrics in an agile development environment. Heimann is a member of the IEEE Technical Working Group that has developed the new 2014 version of the IEEE Standard 730 for Software Quality Assurance. He can be reached by email at heimann.david@gmail.com.

## **APPENDIX** Processes Related to the Software Life Cycle

**I. SOFTWARE SPECIFIC PROCESSES**

**A. Software specific processes**

1. Software implementation process
2. Software requirements analysis process
3. Software architectural design process
4. Software detailed design process
5. Software construction process
6. Software integration process
7. Software qualification testing process

**B. Software support processes**

1. Software documentation management process
2. Software configuration management process
3. Software quality assurance process
4. Software verification process
5. Software validation process
6. Software review process
7. Software audit process
8. Software problem resolution process

**C. Software reuse processes**

1. Domain engineering process
2. Reuse asset management process
3. Reuse program management process

**II. SYSTEM CONTEXT PROCESSES**

**A. Agreement processes**

1. Acquisition process
2. Supply process

**B. Organizational project-enabling processes**

1. Life-cycle model management process
2. Information management process
3. Project portfolio management process
4. Human resources management process
5. Quality management process

**C. Project processes**

1. Project planning process
2. Project assessment and control process
3. Decision management process
4. Risk management process
5. Configuration management process
6. Information management process
7. Measurement process

**D. Technical processes**

1. Stakeholder requirements definition process
2. System requirements analysis process
3. System architectural design process
4. Implementation process
5. System integration process
6. System qualification testing process
7. Software installation process
8. Software acceptance support process
9. Software operation process
10. Software maintenance process
11. Software disposal process