Guidelines for Testing, Verification & Validation of System Requirements

Automation systems are ubiquitous across many industries. Such systems comprise, for example, basic anti-collision systems for manual-controlled cranes, sophisticated autonomous robotic depalletizing systems, warehouse management systems, highly regulated nuclear applications, and many more.

Irrespective of the level of complexity, there are best-practice engineering methods that apply to all automation systems. Of particular import are those methods related to ensuring that a given end system meets the functional requirements for which the system is intended.

It is common for many system functions to be inextricably linked to system software. For these types of systems, industry standards exist that help to ensure that a minimum level of quality is met during the engineering process. The quality process for software engineering is typically referred to as Software Verification & Validation (V&V).

The objectives of this document are to provide an overview of industry standards that are most relevant to V&V, and also to describe the pragmatic implications of these standards for testing, verifying, and validating system functional requirements.

ISO 9001 & ISO 90003

ISO 9001 - Quality Management Systems Requirements - is used by organizations to create policies, processes, and procedures to provide products and services that meet customer needs, are compliant with regulatory constraints, and improve customer satisfaction. ISO 9001 is intentionally constructed as a generic standard so that the standard can be applied to any organization, any sector of business activity, and any product or service.

Since ISO 9001 is a generic standard, the application of this standard to a specific product is aided by other industry standards. For example, application of ISO 9001 to a software-based automation systems is aided by ISO 90003 - Guidelines for the Application of ISO 9001 to Computer Software. This standard recommends the application of a third standard: ISO 12207 - Systems and Software Engineering - in order to comply with the requirements of ISO 9001. A corresponding standard to ISO 12207 is IEEE 1012 - Standard for Software Verification & Validation.

IEEE 1012 & ISO 12207

The IEEE 1012 standard is one of the most recognized industry standards for the V&V process. The standard is broadly applicable to automation systems because the content of the standard is directed at software-based systems, computer software, hardware that is controlled with software, software interactions with operators, and interfaces between software components, among other things.

Broadly, IEEE 1012 sets forth structured methodologies for 1) developing quality software, 2) verifying software compliance with quality standards, and 3) demonstrating objectively whether a given system satisfies its intended use and user needs.
These objectives are achieved through various V&V tasks that are executed during different *phases*[1]

[1] The word used for *phases* within the IEEE 1012 standard is *activities*.

of a project. For example, the V&V task of *Generating an Integration Test Plan* occurs during the *Design* phase of a project. Execution of the Integration Test Plan occurs during the *Test* phase of the project.

It can be noted that the requirements prescribed by IEEE 1012 correspond to equivalent requirements prescribed by ISO 12207. The interested reader may find a mapping of phases and tasks between these two standards in Annex A of IEEE 1012.

Software Integrity Level

Not every V&V task described in IEEE 1012 is executed on a given project. Tasks are prescribed in the standard on the basis of a metric that quantifies the criticality of a given system. The level of criticality represents the complexity, risk, safety level, desired performance, reliability, or other system attributes that define the importance of the system to the end user.

The criticality metric is referred to in IEEE 1012 as the *Software Integrity Level*. Systems which are judged to have a higher software integrity level are prescribed more V&V tasks than those with a lower software integrity level. This results in more rigorous application of V&V tasks for systems that have a high software integrity level. The following table is an excerpt from IEEE 1012 that provides guidance for determining an appropriate software integrity level. Ordinarily, a system as a whole is assigned the same integrity level as the highest level assigned to any individual element within the system.

Description	Level
Software element must execute correctly or grave consequences (loss of life, loss of system, economic or social loss) will occur. No mitigation is possible.	4
Software element must execute correctly or the intended use (mission) of the system/software will not be realized, causing serious consequences (permanent injury, major system degradation, economic or social impact). Partial to complete mitigation is possible.	3
Software element must execute correctly or an intended function will not be realized, causing minor consequences. Complete mitigation possible.	2
Software element must execute correctly or intended function will not be realized, causing negligible consequences. Mitigation not required.	1

Pragmatic Implications of the V&V Process

Among the many phases and tasks prescribed in IEEE 1012 are those that specifically address testing, and the functional requirements to which the testing is directed. These tasks include, for example: Reviewing system requirements for consistency with user requirements; Tracing functional requirements to various system elements to verify that all system requirements are targeted for fulfilment; Performing an interface analysis to verify that all interfaces are described, defined, and testable; Generating & executing V&V test plans that address all functional requirements; Generating & executing acceptance test plans that address customer acceptance criteria; and many others.

The V&V tasks mandated by IEEE 1012 have meaningful implications for system engineering, system requirements, and system testing. Succinctly, and in plain language:

1. IEEE 1012 places extreme emphasis on the process of developing requirements. They must be internally consistent, consistent with customer needs, traceable, and complete. The reason for this emphasis is because requirements, which are derived from the needs of the customer/user, are the foundation of design, implementation, and testing activities.

   There are different hierarchies of requirements. At the highest levels are those requirements imposed on the system as a whole. Since the system interfaces with external systems, system requirements also include interface requirements that constrain the interfaces between the given system and external systems.

   Lower-level requirements derive from the system requirements. Examples of lower-level requirements include integration requirements, component requirements, software requirements, and interface requirements that are imposed on interfaces internal to the given system.

2. IEEE 1012 prescribes that several different categories of tests shall be executed. These include tests directed at the component level, the integration level, the system level, and finally those directed toward acceptance of the system by the customer.

3. Test execution of the various test categories is not adhoc. Instead, IEEE 1012 prescribes careful derivation of test procedures from test plans. Importantly, test plans in each category shall be derived from component requirements, integration requirements, system requirements, and acceptance requirements, respectively. In this way, testing activities are directly traceable to requirements.

4. Test procedures shall include tests that validate compliance with each and every functional requirement: component tests shall validate all component requirements; integration tests shall validate all integration requirements; system tests shall validate all system requirements; and acceptance tests shall validate all acceptance requirements. Additionally, these tests shall include activates that validate compliance with each and every interface requirement.

5. Test plans shall demonstrate the feasibility/capability of the system to be operated and maintained in accordance with the user needs. This requirement emphasizes the importance of enabling a customer to use their system pragmatically in an operational setting. Accordingly, IEEE 1012 mandates that vendor testing shall validate this capability.

6. The results of all tests shall be assessed against acceptance criteria. Acceptance criteria is derived from requirements in each category of test.

7. Interfaces, including those separating the given system from external systems, shall be fully defined and documented. All information required to fully define the interface shall be included in the interface description. This can include information such as data format, timing, and performance criteria. Moreover, objective acceptance criteria for validating interfaces, including those separating the given system from external systems, shall be developed and used during testing to identify interface anomalies.

   In short, all interfaces should be defined and tested against objective criteria to identify anomalies.

8. The contents of the acceptance test plan shall be the prerogative of the acquirer. This means that the acceptance test activities, and the acceptance criteria for whether or not those activities pass or fail the tests are determined by the acquirer.

9. There shall be objective acceptance criteria for all design elements of the system. Moreover, the design elements of the system shall be tested against the objective acceptance criteria.

10. The software components of the system shall be tested to satisfy the system requirements.

11. The software ultimately installed for use by the end user shall be the software that was subjected to the V&V activities.

REFERENCES

[1] International Organization for Standardization. *Quality management systems – requirements*. ISO 9001:2015. Available from https://www.iso.org/standard/62085.html

[2] International Organization for Standardization. *Guidelines for the application of ISO 9001 to computer software*. ISO 90003:2014. Available from https://www.iso.org/standard/66240.html

[3] International Organization for Standardization. *Systems and software engineering – Software life cycle processes*. ISO 12207:2008. Available from https://www.iso.org/standard/43447.html

[4] Institute of Electrical and Electronics Engineers. Standard for System, Software, and Hardware Verification and Validation. IEEE 1012. Available from https://standards.ieee.org/ieee/1012/7324/