Independent Verification & Validation Processes

1 REQUIREMENTS DEVELOPMENT

Our strategy for all software development projects is the evolutionary strategy. This determines user needs and defines system requirements based on available information. The system is built based on these established requirements, along with any currently existing customer requirements, if any. As more information becomes available, the project evolves to encompass the new requirements.

The development process includes the activities for requirements analysis, design, coding, integration, testing and installation and acceptance.

The purpose of the information generated during the development process is to:

- Describe and record information about the software product during its lifecycle.
- Assist in the usability and maintainability of the software product.
- Define and control life cycle processes.
- Communicate information about the system, software product or service and the project to those who need the information.
- Provide a history of what happened during development and maintenance of the software product to support management and process improvement.
- Provide evidence that the processes were followed.

The following activities must be completed for conformance to the development process:

- Process implementation
- System requirements analysis
- System architectural design
- Software requirements analysis
- Software design
  - Architectural design
  - Detailed design
- Software coding
- Software code reads
- Software unit tests
- Software integration tests
- Software system tests
- Software installation
- Software acceptance support

The development life cycle process consists of documentation, configuration, management, problem resolution, and supporting processes.

2 TOOL VALIDATION/QUALIFICATION

Any tool used to help with testing must be qualified as follows:

Section 12.2 of RTCA/DO-178B states that qualification of a tool is needed when processes in RTCA/DO-178B "are eliminated, reduced, or automated by the use of a software tool, without its output being verified as specified in section 6" of RTCA/DO-178B. RTCA/DO-178B states, "The objective of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced, or automated." (FAA 2003)

3 TEST PLANNING AND EXECUTION

System Level Testing

System level testing demonstrates that all specified functionality exists, and that the software product is trustworthy. This testing verifies the program's functionality and performance with respect to the requirements for the software product as exhibited on the specified operating platform(s).

System level software testing addresses functional concerns and the following elements of a device's software that are related to the intended use(s):

- Performance issues (e.g., response times, reliability measurements)
- Responses to stress conditions (e.g., behavior under maximum load, continuous use)
- Operation of internal and external security features
- Effectiveness of recovery procedures, including disaster recovery
- Usability
- Compatibility with other software products
- Behavior in each of the defined hardware configurations
- Accuracy of documentation

Regression Testing

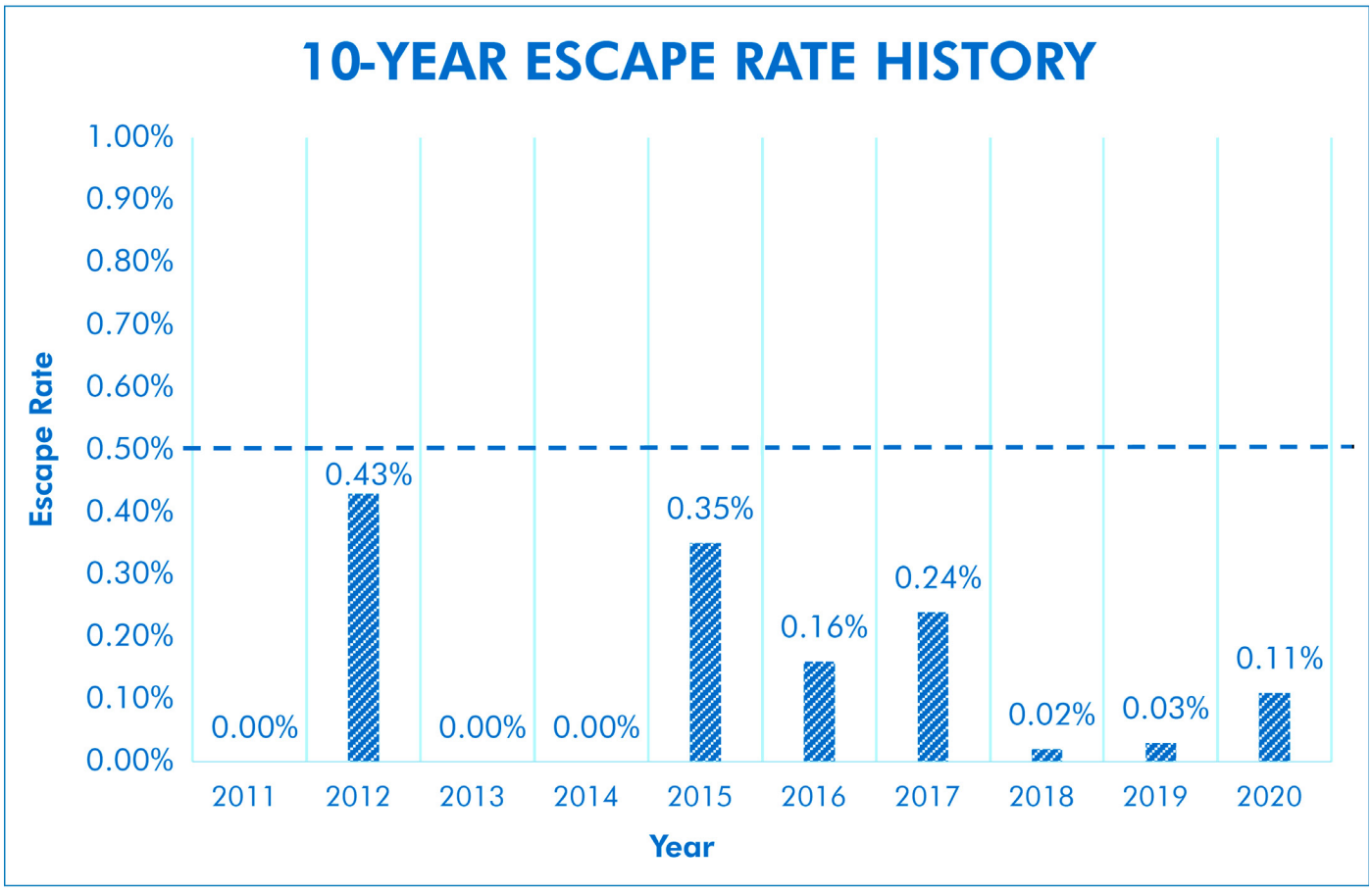
Regression analysis and testing are employed to provide assurance that a change has not created problems elsewhere in the software product.

Regression analysis is the determination of the impact of a change based on review of the relevant documentation (e.g., software requirements specification, software design specification, source code, test plans, test cases, test scripts, etc.) to identify the necessary regression tests to be run. Regression testing is the rerunning of test cases that a program has previously executed correctly and comparing the current result to the previous result to detect unintended effects of a software change.

Regression analysis and regression testing are also employed when using integration methods to build a software product to ensure that newly integrated modules do not adversely impact the operation of previously integrated modules.

Functional tests are written from a user's perspective. These tests confirm that the system does what users are expecting it to do. This is also called **Software Validation.**

Testing an application for the modification of any module or functionality is termed as **Regression Testing.** It is difficult to cover all the systems in Regression Testing, so typically Automation Testing Tools are used for these types of testing.



The dotted line represents General Digital's ISO 9001:2015 goal of maintaining less than 0.5% escapes annually.

Integration Testing (HSIT, SSIT)

Integration level testing focuses on the transfer of data and control across a program's internal and external interfaces. External interfaces are those with other software (including operating system software), system hardware, and the users and can be described as communications links.

Testing of all integrated modules to verify the combined functionality after integration is termed as **Integration Testing.**

Modules are typically code modules, individual applications, client, and server applications on a network, etc. This type of testing is especially relevant to the client/server and distributed systems.

Unit Testing for Safety-critical Embedded Systems

Unit (module or component) level testing focuses on the early examination of sub-program functionality and ensures that functionality not visible at the system level is examined by testing. Unit testing ensures that quality software units are furnished for integration into the finished software product.

Unit tests tell the engineer that the code is doing things right. They ensure that a function/subroutine successfully performs a set of specific tasks. Each test confirms that a function/subroutine produces the expected output when given a known input.

Unit tests also provide a method of verifying that code is complete.

Testing of an individual software component or module is termed as **Unit Testing.**

It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.

Prior to the implementation of FAA IV&V standards, we understood the need for improved traceability and process optimization to ensure that code can be tested in a way that is fast, safe, and reportable. Our engineers developed proprietary unit testing software that has been iterated and augmented since its inception in 1998 in adherence to the ISO 9001:2015 QMS principle of continuous improvement.

Our unit test tool suite, **GenIViVe™**, predetermines many of the necessary conditions to meet the requirements of certain constructs and automates tracking, reducing manual steps and letting our testers do what they do best—build and execute test cases. This process saves time and money, standardizes quality by reducing user error, and provides submission-ready reporting. It interfaces with COTS tools including VectorCAST, LDRA, and TRACE32 along with several proprietary customer tools, and can be configured to interface with other tools if required.

Internal Peer Review

Internal peer review is intended to subject the unit test work product to a structured quality control procedure to ascertain the reliability and supportability of the test.

Reliability is the degree to which the test satisfies the requirements of a unit test. Some questions asked in determining the reliability of the test are:

Was correct boundary analysis performed? Was modified condition/decision coverage performed? Were all outputs and inputs initialized for each test case? Do all expected outputs match the simulator returned outputs? Were all the tests performed on live pins or code? Was overflow tested? Is code coverage complete?

Supportability refers to the construction and documentation of the unit test. The goal of supportability is to be able to refer precisely and easily to which test cases satisfied which test requirements. Furthermore, the test should be usable as a baseline for future tests of new versions of the module. Some questions asked in determining the supportability of the test are: Are there complete and correct decision notes (or test strategy) written for the module?

Are the test cases organized and readable? Is all other documentation, correct? Was the correct version of the module tested and is this version identified in test files and UT form?

To the end of ensuring the reliability and supportability of the unit tests performed by General Digital, 100% of all modules will be subject to the review procedure outlined below. In some cases, an analysis of all constructs and test cases is neither practical nor possible given time and resource constraints. A disciplined sampling methodology outlined in this procedure will therefore be employed in the review to establish confidence in the quality of the test.

This procedure applies to all Software Engineering Services personnel involved with the creation of GD Software Engineering Services work product. This peer review procedure is specific to unit test; however, it may be applied to similar verification activities.

ADDITIONAL TOPICS

Code Reads

The verification process also includes a visual review of the code (code reads). The reviewer completes a code read form for every software item.

Safety/Security Critical Software

Safety/Security Critical software requirements are verified for consistency, feasibility, and testability. Software requirements specifically related to safety and security are given extra scrutiny to help ensure their correctness. This includes documentation detailing the reasoning behind a requirement. This also allows for easier identification of requirements that require closer examination.

Verification is performed by Independent Verification and Validation engineers. Safety/security critical code is then reviewed by a qualified reviewer other than the person who wrote the code. Formalized functional testing and unit testing is also performed on such code.

4 PROJECT COORDINATION

All job functions and workflow items are subject to a General Digital developed proprietary software tool called **GenTrack™**. **GenTrack** is designed to integrate General Digital Software Engineering Services (GD-SES) processes of project management, timekeeping, and invoicing functions for the Verification and Validation (V&V) group into a closed-loop system.

The primary function of the system is to link the engineer timesheet form to the V&V tracking forms generated for the customer. In this way, each hour submitted on a timesheet will be accounted for in invoice-ready V&V and project management tasks, or an internal GD-SES project.

Secondary functions of this system are to:

- Provide accurate and verified input to the business office in a timely manner.
- Digitize and centralize workflow information within tables to remove paperwork errors.
- Provide Controllers with better methods to analyze and update workload information and to analyze employee performance data.
- Provide security and back-up capabilities for the data.
- Ensure specific tasks and workflow events (i.e., overtime) are authorized.

5 DISCREPANCY RESOLUTION

All non-conforming work product and customer complaints are subject to review and rectification by qualified personnel. GD uses a Corrective Action Request (CAR) document to rectify non-conformities.

Once the need for a corrective action has been identified, anyone within GD-SES can initiate a corrective action request. Corrective actions are uniquely identified to facilitate tracking.

GD also has a Preventive Action Request (PAR) document that is required where a potential for generating non-conforming work product is identified or where a procedure or work instruction requires updating for ease of use and/or increased quality.

A PAR is generated when a deficiency which could lead to non-conforming work product is identified. It applies to all GD-SES activities and is used to eliminate potential non-conformances.

6 RESULTS REPORTING

Over the course of the lifecycle, the IV&V Team may generate analysis reports that document the results of the analyses performed. These reports will typically describe what the IV&V Team analyzed (project artifacts), a high-level description of the process, approach, and tools used (if applicable), and associated results.

GENERAL DIGITAL
An SBA Small Business Concern
60 Prestige Park Road
East Hartford, Connecticut 06108
Phone 860.282.2900 **Toll-Free** 800.952.2535
E-mail gdc_info@generaldigital.com
Web www.generaldigital.com

ISO 9001:2015 QUALITY MANAGEMENT SYSTEM CERTIFIED

999-0908-001r1