Independent Test Verification: Consolidated Experience Report

Nuno Silva[1], Rui Lopes[2]

[1]Project Management Office – ASD, Critical Software, S.A., Parque Industrial de Taveiro, Lote 48, 3045-504 Coimbra, Portugal

[2]Aeronautics, Space and Defense, Critical Software, S.A., Parque Industrial de Taveiro, Lote 48, 3045-504 Coimbra, Portugal

{nsilva,rmlopes}@criticalsoftware.com

Abstract. *Independent verification and validation (IV&V) has been a key process for decades, and is highlighted in several international certification standards. One of the activities described in the "ESA ISVV Guide" is independent test verification (stated as Integration/Unit Test Procedures and Test Data Verification). This activity is commonly overlooked since customers do not really see the added value of checking thoroughly the validation team work. This article presents the consolidated results of a large set of independent test verifications, including the main difficulties, results obtained and advantages/disadvantages for the industry of these activities. This study will support customers in opting-in or opting-out for this task in future IVV contracts since we provide factual results from some real case studies.*

1. Introduction

Software Quality has long been a common goal to the systems/software industry. Some have mastered the "art" of producing high quality software, some are still trying to acquire the basics of software quality and working hard to convince stakeholders that software needs high quality, appropriate processes, careful maintenance, and above all, as much dependability as it can have.

One of the trends for guaranteeing maximum software quality is IV&V and its application to software (Independent Software Verification and Validation - ISVV). The increasing complexity, size and importance of the software lead to an increasing demand on IV&V since several decades ago. This demand is directly related to the establishment of IV&V activities as mandatory in several international standards. IV&V got developed and consolidated being today widely used by organizations such as National Aeronautics Space Administration (NASA), European Space Agency (ESA), Department of Defense (DoD) and Federal Aviation Administration (FAA), amongst many other important international institutions. IV&V is also mentioned in international standards such as ISO/IEC 12207 [ISO 2008] and IEEE 1012 [IEEE 2004]. In Europe, "ESA ISVV Guide" [ESA 2008] was developed by a consortium lead by ESA, including Critical Software. This guide consolidated the ISVV process for safety-critical systems.

Following a SDP is an essential step in order to achieve software with quality and to control all the steps of software production. Software is more and more under

pressure, thus software quality issues and complaints are more frequent and the failures have more severe consequences.

Software Quality has always been the whole of the quality of software production artifacts. It depends on specifications/requirements quality, architectural and design quality, source code quality, verification and validation quality and care, etc. All artifacts have their importance and the earlier quality is controlled and imposed the better the product will be developed. This being said, the testing phase cannot be ignored: it's an essential phase to avoid problems from remaining in the system, to validate and provide confidence on the system, to detect issues but also to confirm that all appropriate artifacts are produced, traced up to system level requirements, fulfilling the system expectations (and sometimes beyond). This industrial experience report is touching this testing phase as one of the most important phases to build/guarantee software quality. The testing (sometimes confused with validation) importance is undeniable, in most industries the validation (includes testing) phase incurs for about 40% (Refer to the 40-20-40 rule for Specification-Implementation-Test as described in [Pressman 2001]) of the time/effort/cost of a system development.

Studies even consider this phase the responsible for the failure of systems [Jones 2005], providing us ground to invest effort in studying the effectiveness of testing.

As one the ISVV/IV&V tasks, Independent Test Verification (ITV) becomes then an important activity, allowing the verification of large amounts of test data, confirmation of the quality of tests performed by a subcontractor and for sets of tests that require independent verification (as required in international standards such as DO-178B [RTCA 92], EN-50128 [CENELEC 2011] or the Galileo Software Standard [GSWS 2004]). This experience report will highlight the importance of these activities (ITV), their added value with some quantitative results and will demystify the idea that errors are only found in requirements, design and code, and the original functional testing activities.

This article includes a short description the independent test verification (section 2), an overview of the industrial case studies that have been used to extract the ITV results (section 3), some of the results obtained (section 4) and a set of derived conclusions concerning the application of these activities associated to safety critical systems development and their contribution to better software quality (sections 5 and 6).

2. Independent Test Verification

Several international standards related to software development processes and software quality for safety-critical systems mention the importance of independence (e.g. [ESA 2008]). Independence has been seen as key to ensure software dependability, since it provides an unbiased assessment of the system under analysis. For the different standards that mention independence, the selection of activities that require some degree of independence might differ.

The ITV activities are defined in section 7 of the "ESA ISVV Guide" [ESA 2008] and they intend to ensure complete and correct Unit/Integration testing. This is due to the fact that the ESA SDP, contrary to e.g. DO-178B, is still not enforcing independency in the definition and execution of the validation activities. Thus, the

validation activities can be carried out by the software (SW) development team, significantly reducing costs with respect to the DO-178B approach. This is why the Test Verification activity has been defined and made mandatory for systems that require ISVV. ITV may optionally extend to the verification of functional/validation tests providing an independent check of the requirements coverage of the original test campaigns.

ITV tasks include traceability verification between the test cases/procedures and artifacts from previous phases of the SDP (specifications, design, and source code), verification of the completeness and correctness of the test specifications and procedures produced by the validation team, verification of the test reports/logs resulting from the unit/integration or functional tests already executed.

The added value of ITV is not consensual; some stakeholders actually believe that ITV effort could be used for improving the requirements verification or (mainly) reinforcing the source code analysis. However, due to the importance and weight of the testing activities, and due to the role that the outputs of the validation activities play in the whole system acceptance and provided quality confidence, these activities can be used as a watchdog for double checking such important activities. This article presents important results that reinforce the relevancy and importance of the independent test verification for safety critical systems.

3. Industrial Case Studies

The case studies used to collect data for this paper are related to the aerospace domain. All these systems are highly critical (a failure in these systems will have a severe impact on the overall system). In such case, the ITV activity has been required by all costumers, in order to increase the confidence in the systems being developed and ensure that appropriate, complete and correct tests have been performed.

The systems included in this study have different characteristics, requirements, and have been developed and tested by different companies. The development and validation of these systems, however, followed mature processes (aerospace industry) and had high quality assurance standards. The level of details of the requirements, design, coding standards followed and relevant validation experience and tools differ from system to system, as well as from development entity to development entity.

All case studies have been developed by companies with high experience in the development of safety-critical systems (all have more than 10 years in this field), with mature SDP and strong orientation towards safety-critical standards compliance.

The complexity and size of these SW modules varies depending on their function in the system, ranging from low-level functions such as boot software or APIs to data handling or attitude control and regular payload application SW. The smallest SW module chosen for the case study for this paper is defined by only 48 requirements and implemented in approximately 3,500 lines of code while the largest has 3,324 requirements and is implemented in approximately 450,000 lines of code.

The findings (issues) derived from the ITV activities have been categorized according to their impact on the system (the severity of the end effect):

Table 1. Severity categories used to classify issues

Category	Description
Comment	The discrepancy found does not present any threat to the system. The issue was raised as a recommendation that aims at improving the quality of the affected item.
Minor	The discrepancy found is a minor issue. Although it does not present a major threat to the system, its correction should be done.
Major	The discrepancy found refers to the lack of pertinent information or presents a threat to the system. The correction and/or clarification of the discrepancy are pertinent.

Not only test related issues (i.e. test correctness or completeness issues) are identified. Traceability between test data and artifacts from previous SW development phases (i.e. design components, requirements) is also verified, often raising questions related to those artifacts and revealing weaknesses in different phases.

The results presented hereafter mainly focus on the relations between the number of issues found and the number of SW artifacts of each project, in order to establish a statistical base to evaluate the impact and the added value that the activity provides to the projects. Revealing concrete problems is not allowed by the customers.

4. ITV Results

The ITV activities have been applied by expert engineers in IV&V and applied to separate systems and subsystems with different sizes and complexities. The following table presents a few properties in order to depict the analyzed systems.

Table 2. Properties of systems under verification

System	System Properties		
	Number of Requirements	***Code Size***	***Number of Tests***
SYS1	48	3,471	10
SYS2	184	12,219	57
SYS3	134	2,545	79
SYS4	831	38,455	130
SYS5	156	17,235	146
SYS6	368	21,783	150
SYS7	691	69,910	156
SYS8	165	4,751	165
SYS9	45	18,876	212
SYS10	3,324	450,778	256
SYS11	88	21,856	42
TOTAL	**6,034**	**661,879**	**1,403**

The column of Requirements represents the software requirements associated with the system or subsystem under study; the column Code Size represents the number of physical lines of code (in either C, Assembly or Ada languages) excluding the lines of

comments; and the column of Tests indicates the number of main test procedures analyzed. These procedures are usually quite complex and each procedure contains several steps (sometimes 20-50 steps).

The ITV activities allowed the identification of a large amount of findings (over the 1,400 tests analyzed and a few thousands of documentation pages associated). Table 3 presents the summary of the issues found and improvements suggested. Note that the acceptance rate of these issues is more than 75%.

Table 3. Overview of verification activities results

System	Comments	Minor	Major	Total
SYS1	0	2	0	2
SYS2	3	6	1	10
SYS3	2	5	3	10
SYS4	0	10	0	10
SYS5	1	4	16	21
SYS6	1	30	8	39
SYS7	3	36	31	70
SYS8	0	0	6	6
SYS9	3	5	6	14
SYS10	24	266	43	333
SYS11	3	14	3	20
TOTAL	**40**	**378**	**117**	**535**

Most of the Comments are usually discarded due to their low importance. Overall, the acceptance rate is high since most of the issues have very strong arguments either on the test artifacts or on the related documentation (that is sometimes outdated).

Similar issues have been identified in several of the systems. One can highlight two that have been detected in all of the systems: requirements not covered and requirements incorrectly covered. The first group of issues is related to requirements that are not covered by the test campaign and no justification for that matter is provided. The second group is mostly related with complex requirements, in which the tests do not cover the full range of functionalities described by the requirement.

The spent effort for the ITV activities that lead to the issues presented in Table 3 was about 2,300 man*hours, including technical management and reporting effort. Some interesting metrics computed from this activity are shown in Table 4.

Table 4. Independent test verification metrics

Metric	Value
Issue/Test	0.38
Issue/Requirement	0.09
Issue/1,000 lines of code	0.81
Test/Hour	0.63

Metric	Value
Issue/Hour	0.24
% Major Issues	22%
Major Issues/Test	0.08

5. Results Observations

This section intends to present the important management and technical issues as well as to discuss the results presented in the previous section, while mapping those results to our conclusions. This work presents relevant arguments for future analysis and for software quality evaluation as a whole.

5.1. Management discussion

From a management perspective, these activities had to overcome several challenges. Both technical and project management are key to the success of any activity. Most of these independent tasks are required to be performed in a very short time-frame, thus increasing the number of resources working with technical coordination, and requiring a specific and consolidated expertise. Resources must be mature and experienced, able to work properly in teams, share their work and findings and integrating the issues. The communication of the issues is always a sensitive question as well as the peer reviews.

Whenever a company is performing this type of assessment, the objective must be to help the customer/supplier of the test artifacts in improving their work and contribute positively to a better system. It's not worthwhile to just criticize the work and point out simply "bugs", the most interesting part of this work is the alternatives and suggestions provides for resolution of the issues.

Management had to deal with internal team monitoring, coordination, integrations and reviews, but also with external communications, collection of all artifacts required by the team to perform the analysis, requests for clarifications, sometimes limited amount of documentation and no technical contact with the tests supplier as requested by the independence requirements of the activities.

5.2. Technical discussion

Section 4 presented some metrics extracted and computed from the ITV activities. The systems (and subsystems) under analysis belong to different projects and have different requirements; they also have different levels of documentation and different coding and testing styles/tools. They have, however, similar quality objectives and similar strict development processes and requirements to follow. They have also been developed by institutions with a large experience in the domain, thus with a similar maturity (more than 10 years in the aerospace software development field). Table 1 represents a few properties that classify the systems from the less complex to the more complex ones.

Table 2 provides the number and severity of issues raised by the IV&V teams. The major issues are the most relevant ones because they represent either severe problems in the testing artifacts or severe problems related to requirements, design or implementation of the system or inconsistencies between the tests and other artifacts.

It is worth mentioning that even if all issues are resolved one can never ensure that the system will have zero defects because the correction of these issues might uncover additional issues or create them, and it is also impossible to guarantee that the independent team can catch all the issues – although it is catching a lot of them.

From the results we can identify a few systems with a larger number of issues (especially Major issues), in this case we can identify SYS2, SYS8, SYS7, SYS6 and SYS10 as the more severe cases, these systems are also the more complex from requirements and source code size perspective. They also represent the highest rates issues/test. Based on this experiment we can conclude that systems 2 and 8 are the ones that need more improvements in what concerns quality and correction of bugs.

Table 4 from the past section presented some metrics useful for ROI calculation and that show the importance of ITV activities. Based on those metrics we can extrapolate the following:

- On average, per 100 requirements we can expect to find 9 issues by ITV;
- On a system with 100 tests we can expect to find 38 issues by performing ITV;
- It will cost about 150 hours to review 100 tests;
- An engineer will find 1 issue per each 4 hours of work;
- For a 40 hour week, 10 issues can be found, of which, 2 issues will be major.

For example, finding 9 issues per 100 requirements appears to be a high value for a system that has already passed all the phases, up to the testing phase. If, at the testing/validation phase we still find 9 discrepancies, the system is still a bit unstable, however, these issues might be small details and inconsistencies due to documentation updates and maintenance issues. This suggests that the ITV activity is well worthwhile. Another example is that of spending 40 hours of ITV and finding 10 issues on average, 2 of them being major issues. If you can still find 2 major issues per week at validation phase you can obviously claim that the system needs improvements and the issue-finding activity (ITV) is providing an excellent return. You might decide on continue dong ITV even once the issues of the first round have been resolved.

6. Conclusions

The number of issues found is unexpectedly high, given the maturity of the SW development teams and the development processes involved. This does not mean that the SW quality was lower than usual, but it highlights the added value that this activity provides to projects. From the issues found, there is a considerable percentage of “major issues” (22%). This corroborates the idea that the activity provides high added value to the projects in which it is applied.

The main qualitative conclusions drawn from the statistical study performed are summarized below:

- The activity is efficient in providing an unbiased assessment of the test coverage;
- Critical issues are often identified with high impact on the system (sometimes going back to specification issues);

- The number of critical issues found is higher when the activity is extended to the verification of the functional tests, increasing the added value of the activity (The findings related to the verification of Unit and Integration tests are generally of lower criticality and with reduced added value).

The verification of functional tests and requirements coverage is highly recommended since most of the added value has been found to be associated to this task.

The ITV allows a significant risk reduction concerning the system's dependability level, and when these activities are performed by expert teams, focused on finding and proposing solutions and suggestion for the issues identified, the results really do provide additional confidence in the system's quality for all stakeholders.

7. Acknowledgement

This work has been partially supported by the project CRITICAL Software Technology for an Evolutionary Partnership (CRITICAL-STEP, http://www.critical-step.eu), Marie Curie Industry-Academia Partnerships and Pathways (IAPP) number 230672, within the context of the EU Seventh Framework Programme (FP7).

8. References

[CENELEC 2011] CENELEC EN 50128: Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems.

[ESA 2008] ESA ISVV Guide, issue 2.0, 29/12/2008, European Space Agency.

[GSWS 2004] GSWS: Galileo Software Standard, GAL-SPE-GLI-SYST-A/0092, 2004.

[IEEE 2004] IEEE 1012-2004 - IEEE Standard for Software Verification and Validation. IEEE Computer Society.

[ISO 2008] ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.

[ISVV 2012] ISVV Web site, www.isvv.com, visited on 13/03/2012.

[Jones 2005] Software Engineering: Are we getting better at it?, Michael Jones, ESA Bulletin 121, February 2005, pp. 52-57.

[Pressman 2001] Software Engineering: A Practitioner's Approach., Pressman, R., McGraw-Hill, 5th Edition, November 2001.

[RTCA 92] DO-178B/ED-12B, Software Considerations in Airborne Systems and Equipment Certification, 01/12/1992, RTCA.