Evoting SIG

Association for Software Testing

Contact: Jon Hagar, Chair
POB 24
Hot Sulphur Springs Co, 80451
Jon.d.hagar@lmco.com
Sept. 27, 2009

U.S. ELECTION ASSISTANCE COMMISSION
1225 New York Ave. NW – Suite 1100
Washington, DC 20005

Dear sirs,

This letter is just a hardcopy confirmation of the inputs we provided via the official electronic input tool. No action based on this letter or hardcopy inputs is required as this is just a courtesy to you.

Any question or follow up you have may be directed to our chair listed above. We look forward to continuing to work with the EAC.

Yours,

s/ Jon Hagar

AST, Chair, eVoting.

Comments from the

Association of Software Testing

on the Election Assistance Commission's

Draft Voluntary Voting System Guidelines

Version 1.1

September 28, 2009

For the Association for Software Testing Special Interest Group on Electronic Voting

Alan A. Jorgensen, Ph.D.
Consulting Software Test Engineer
"Over 50 years of testing experience."

Jon Hagar
Software Test Professional

James Nilius, CTFL
Software Test Architect

Reid Sheppard
Manager Of Software Test Engineering

Table of Contents

Testing of Commercial Off The Shelf (COTS) Software as Applied to Electronic Voting Systems..... 3

Comments on Volume I Section 5.2 5

References from Section 6.4.1 (2007)..... 5

Volume I Section 5.3.5 (From 6.4.1.5-B.3) Unstructured Exception Handling..... 5

Volume I 5.2.7 a (ii) (from 6.4.1.7-A.2) Unsafe concurrency 5

Volume I 5.2.8 a. (from 6.4.1.8-A.1) Defense against garbage input..... 6

Volume I 5.2.8 k. (from 6.4.1.8-K) Election integrity monitoring 6

Testability 6

Comments on Volume II Section 1.8.2.6 7

Conformity assessment is not quality assurance. 7

Logic Defects responsible for incorrect recording, tabulation, or reporting of a vote 7

Test Plan update and approval for non-terminal logic errors regression test updates..... 8

Comments on Appendices A and B of Volume II 9

Volume II Appendix A (2005 V1.1) Section A.2 Required Content of Test Plan 9

Volume II Appendix A (2005 V1.1) Section A.3.3 Software Module Test Case Design and Data 9

Ensure Manufacturer module test case design provides conclusive coverage 10

Explicit statement of Manufacturers module test coverage requirement 10

Path coverage is an impossible requirement 11

Volume II Appendix B (2005 V1.1) National Certification Test Report 12

Testing of Commercial Off The Shelf (COTS) Software as Applied to Electronic Voting Systems

Jon Hagar, AST eVoting Special Interest Group Lead
Alan A. Jorgensen, Ph.D., Consulting Computer Software Test Engineer

This short white paper is written without reference information, but references can be provided if requested.

We have been involved in Commercial Off The Shelf (COTS) software testing for many years. COTS based systems are standard in the commercial software industry. Systems based on COTS software, software as a service, service orient architectures, and the like, are in common use. In all cases that we know of, the functional and non-functional requirements fulfilled by COTS software requires some level of verification and validation, at least at the system's black box level, as used in the system in which it is included. In Information Technology, government applications, and commercial usage, COTS software must be tested at a base, integrated, and system level. Plans should be prepared defining the testing of the COTS software at element level to ensure that that COTS software is providing the functionality specified by the system vendor in the form of an acceptance test. This test should determine the applicability of that COTS software to the application under development. Failure of any test indicates that the end product using that COTS software would also fail. Blind verification acceptance of COTS features is not an accepted practice.

Additionally, system product life cycle planning should address the evolution and include plans for the end of life of the COTS elements as well as the manner in which updates or replacement versions of the COTS elements become available.

The importance of testing COTS is associated with the software principles of coupling and cohesion. Coupling is the degree to which each program module relies on each one of the other program modules. Software can either be loosely or tightly coupled. Coupling occurs via data transfers and other communication between modules including COTS modules. Cohesion can be thought of as the degree of singleness of purpose of a software component. The application of COTS elements and interfaces should be designed to maximize cohesion via the use of things such as glue-ware and wrapper technologies; however no technology can completely prevent COTS side effects from introducing problems into a system because of the principle of coupling. When developing and fielding COTS based systems we find it necessary to follow the Russian proverb "Trust, but Verify". This should be particularly true of any COTS element and system that is being used in a mission critical domain such as eVoting since the likelihood of first-of-a-kind issues and interfaces is greater. COTS functionally, even when proven in one environment and application, can, and will, perform differently when it is used in a different context.

Most modern systems are built using COTS elements, as mentioned above, and whole classes of development, e.g. Search Oriented Architectures (SOA), have evolved around them. But SOA and COTS IT systems in practice and in published literature require test plans, facilities and active targeted testing of the COTS element. This lesson has been learned in a variety of domains and market places including

government IT, DoD, FAA, NASA, and others. The fact that the VVSG is very weak (to non-existent) in the COTS testing and evaluation areas should be corrected to include specific additions of the following types of requirements:

1. Production of test plans by vendors and independent testers that address COTS testing.
2. Functional and Nonfunctional test methods and techniques targeting the COTS elements.
3. Required Documentation from the COTS vendors.
4. Documentation of the COTS testing by all testers.
5. Evaluation of the COTS prior to selection via trade studies, requirements allocation, and design elements.
6. COTS based risk identification and testing.
7. Plans for COTS version upgrade, replacement, and end-of-life.
8. Documentation of COTS error handling, patches, and fix methodologies.
9. Production and identification of COTS lessons learned both in VVSG and vendor documentation.
10. Inclusion of the handling of COTS failures in the voting system certification test process.
11. Subjecting the COTS portions of an electronic voting system to the same level of testing rigor as the portions provided by the vendor.

These do not mean every part or line of code of COTS need be tested. Access to COTS code is not a requirement and testing can be conducted without it. COTS elements and interfaces have proven the source of many system failures, and so while their use is expected within eVoting systems, they must be subjected to testing and verification.

Comments on Volume I Section 5.2

References from Section 6.4.1 (2007)

Recommendation: Adding References to Works of Recognized Testing Authors

This is a general recommendation for Volume I Section 5.2. This section has the following statement "use of widely recognized and proven logic design methods will facilitate the analysis and testing of voting system logic" as well as several "reference" types of statements in other parts of text. We recommend references be required to leading authors and industry standards worded as follows after the above reference line - "To meet this, vendors shall define in company and/or project documentation which industry practice standards and authors are being used and if these are general information or compliance, e.g., works by: Peterson, Suzanne Robertson, Cem Kaner (best seller), Rex Black, and Boris Beizer as well as reference to IEEE standards (e.g. 1012, 829, 730, 830, 1008, 12119, and 12207), CMMI and/or ISO 9000 standards."

Volume I Section 5.3.5 (From 6.4.1.5-B.3) Unstructured Exception Handling

Recommendation: Requirement for Prohibition of Unstructured Exception Handling

In Volume I Section 5.3.5 (From 6.4.1.5-B.3), "Unstructured exception handling (e.g., On Error GoTo, setjmp/longjmp, or explicit tests for error conditions after every executable statement) is prohibited."

This statement should have a "SHALL BE" added in place of "is".

Volume I 5.2.7 a (ii) (from 6.4.1.7-A.2) Unsafe concurrency

Recommendation: Verification of Race Conditions, Deadlocks, Livelocks, and Resource Starvation

Volume I Section 5.3.5 (From 6.4.1.5-B.3) Unstructured Exception Handling, states ""Application logic shall be evaluated using multiple industry accepted methods and tools for race....." . It would be better to read, "Application logic shall be evaluated for unsafe concurrency using multiple industry accepted methods and/or tools for race conditions, deadlocks, livelocks, resource starvation and other non-functional (performance) issues. Planning for the evaluations shall be contained in project documentation e.g. software development plans, test plans, systems engineering plans, etc."

Volume I 5.2.8 a. (from 6.4.1.8-A.1) Defense against garbage input

Recommendation: Addition of Specific Tests of Information Input

Volume I 5.2.8 a. states "All programmed devices shall check information inputs for completeness and validity and ensure that incomplete or invalid inputs do not lead to irreversible error." This section should have specific testing requirements as does section 5.2.8 b. that follows. In particular, for example, add paragraph "i. Boundary value testing of all scalar or enumerated type input category partitions must be tested. Categories must include all invalid as well as valid inputs. Information provided to the system by any user including voters and election workers must be test. This applies to inputs of numeric values, character values, and any other types for which the concept of range or scope is well-defined. In addition, the length of every input string, whether provided manually or by a data device, must also be considered a category with appropriate valid and invalid boundary values."

Additionally, the first statement should be reword to "Programmed devices SHALL check information inputs for completeness and validity producing, and permanently recording, error indicators where device information inputs and indicators are defined in the device's project requirements specification document. This is to minimize the chance for irreversible errors and allow traceability/auditability of the vote."

Volume I 5.2.8 k. (from 6.4.1.8-K) Election integrity monitoring

Recommendation: Clarification of Election Integrity Monitoring

Volume I 5.2.8 k. (from 6.4.1.8-K) Election Integrity Monitoring reads "To the extent possible, electronic devices SHALL proactively detect or prevent basic violations of election integrity (e.g., stuffing of the ballot box or the accumulation of negative votes) and alert an election official or administrator if they occur." The words "to the extent possible" needs definition. They are not testable as written. It would read better with the replacement statement, "Electronic devices SHALL proactively detect basic violations of election integrity, and then alerting an election official or administrator if they occur, while maintaining permanent records of the volition where violations at a minimum, include: a. Stuffing of the ballot box; b. Accumulation of negative votes; c. Roll back of counts; d. Roll forward of counts; and e. Accumulation of multiple entries (votes) for a single voter."

Testability

Recommendation: Considerations for Testability of Requirements in Volume I, Section 5.2

In general in Volume I, Section 5.2, we see a lack of "testability". We propose that the definition of workmanship include producing systems that are testable, both by the vendors and secondary parties. Definition of testability to include: 1. definition of inputs and outputs to allow test methods such as equivalence class testing, boundary value analysis, output forcing, nominal and stress cases, etc.; 2.

Support for Usability testing; and 3. Support for functional testing; and 4. Support for non-functional testing.

Comments on Volume II Section 1.8.2.6

Conformity assessment is not quality assurance.

Recommendation: Distinguishing Conformity Assessment and Quality Assurance

Volume II Section 1.8.2.6 of V1.1, Certification Test Practices, second paragraph states "Conformity assessment is not quality assurance. If a critical software defect (a software defect responsible for the incorrect recording, tabulation, or reporting of a vote) is found, the system cannot be considered trustworthy even after the known fault is corrected, because the cases that the VSTL does not have the opportunity to test can be expected to conceal similar faults. Therefore, ..."

We recommend that a statement be added to the standards following the first sentence of this section paragraph that provides additional clarification that the VSTL is NOT responsible for the Quality Assurance and Quality Control of the Voting System.

"The VSTL is only responsible for conformity assessment and verification that the system under test satisfies the applicable standards and meets the requirements for EAC certification. Quality Assurance and Control is the complete domain of the Manufacturer and the VSTL SHALL NOT serve this role when the Manufacturer performs these activities inadequately. If the VSTL so determines that the manufacturer has not performed these activities under full conformance to the Manufacturer's documented internal processes then the system must be rejected and returned to the Manufacturer for the comprehensive completion of those inadequate activities and a new application to the EAC must be submitted."

Logic Defects responsible for incorrect recording, tabulation, or reporting of a vote

Recommendation: Response to Logical Defects

Volume II Section 1.8.2.6.d of V1.1, Certification Test Practices, states "d. If a logic defect is found that is not responsible for the incorrect recording, tabulation, or reporting of a vote, testing shall be suspended and the system returned to the manufacturer for correction and quality assurance. The failure shall be counted in the evaluation of reliability (see Appendix C). Nevertheless, the manufacturer will be given the opportunity to correct noncritical software defects. Revisions to the software must be performed within the manufacturer's quality assurance and configuration management processes and must undergo manufacturer regression testing before the conformity assessment process is resumed. When it is resumed, the test plan should be revised to include regression testing for the change that was made."

We recommend that the last sentence of this paragraph be written to state "..., the test plan SHALL be revised ..."

We recommend that in this instance that a formal review process be initiated to make the determination that such a logic defect has occurred and is of the correct nature as to force the rejection of the system given the costs in monetary, human resources and time. We therefore recommend that a new section be added after Volume II, Section 1.8.2.6.d as follows:

"e. Given the scope and ramifications of the penalty associated with the finding of such a logic error at this point in the conformance assessment, a formal review process shall be initiated by the EAC. They shall bring the Manufacturer and the VSTL together with the EAC Technical Reviewers to analyze the voting system logic defect and make the determination that such a logic error has occurred and is of the correct nature as to force the rejection of the system."

The current paragraphs "e." and "f." will need to be renumbered "f." and "g."

Test Plan update and approval for non-terminal logic errors regression test updates

Recommendation: Timely Review of Regression Test Plan Updates

Volume II Section 1.8.2.6.d of V1.1, Certification Test Practices, states, in part: "When it is resumed, the test plan should be revised to include regression testing for the change that was made."

We recommend that guidelines for the update of the Test Plan be documented and formalized to speed the review and acceptance of the changes to the Test Plan for the regression testing of non-terminal defects.

These criteria must be satisfied by the VSTL to facilitate the review and approval of the Test Plan by the EAC's Technical Reviewer. This is to ensure a timely EAC Test Plan modification approval so that the VSTL's will be able to execute the required additional testing in a timely manner.

Comments on Appendices A and B of Volume II

Volume II Appendix A (2005 V1.1) Section A.2 Required Content of Test Plan

Recommendation: Expand how and where 'Known Field Issue' data is obtained

Volume II Appendix A Section A.2, Required Content of Test Plan, subsection, Pre-Certification of Testing and Issues, states "*Known field issues*. The VSTL shall list relevant issues uncovered during field operations."

There is currently no 'official' clearinghouse for this voting field issue data at this time. In the absence of this official 'source of record', the fulfillment of this requirement cannot be satisfied by the VSTL without some method of providing this data in some form so that it can be used in the analysis and construction of effective test cases for conformance testing.

We recommend that, as there is currently no clearing house for this data, the EAC require that this 'known field issues' data be supplied by the Manufacturer as an additional component to the Technical Data Package.

In Volume II, Section 2.1.1, Content and Format, a statement must be added following sub-point g. as subpoint "h. Manufacturer field issue data for the voting system being submitted for certification testing shall be supplied".

Additionally, in Volume II, Section 2.1.1.1, Required Content for Initial Certification, subsection Technical Data Package, main part, a bullet must be added following bullet 9, that states, "10. Known field issue data for the system".

Finally, to ensure that the same information is provided for system changes, in Volume II, Section 2.1.1.2, Required Content for System Changes and Re-Certification, the second paragraph be modified by changing the MAY to SHALL as follows: "Manufacturers SHALL also submit other information relevant to the evaluation of the system, such as test documentation, and records of the system's known field issues, performance history, failure analysis and corrective actions."

Volume II Appendix A (2005 V1.1) Section A.3.3 Software Module Test Case Design and Data

Recommendation: Eliminate the Module Test Case Design, if available statement

Volume II Appendix A Section A.3.3, Software Module Test Case Design and Data, states "The VSTL shall review the manufacturer's program analysis, documentation, and, if available, module test case design. The VSTL shall evaluate the test cases for each module, with respect to flow control parameters and data on both entry and exit. All discrepancies between the Software Specifications and the test case design shall be corrected by the manufacturer prior to initiation of certification testing."

We recommend that the "*if available*" clause be removed from the standards. The statement should read "The VSTL SHALL review the manufacturer's program analysis, documentation, and module test case design."

This makes those module test case designs available to the VSTL so that they will perform a thorough review of these tests, which cannot be accomplished without those test designs.

Ensure Manufacturer module test case design provides conclusive coverage

Recommendation: Ensure Manufacturer module test case design provides conclusive coverage

Volume II Appendix A Section A.3.3, Software Module Test Case Design and Data, states in the second paragraph "If the manufacturer's module test case design does not provide conclusive coverage of all program paths, then the VSTL shall perform an independent analysis to assess the frequency and consequence of error of the untested paths. The VSTL shall design additional module test cases, as required, to provide coverage of all modules containing untested paths with potential for untrapped errors."

We recommend that the VSTL not be required to perform this level of testing when they deem the voting system module test case design inadequate to provide the coverage that the requirements specify. Instead, the system should be rejected and returned to the Manufacturer if they have not adequately satisfied this requirement of path and error coverage at the module level.

The paragraph should be changed as follows:

> Remove the text "The VSTL shall design additional module test cases, as required, to provide coverage of all modules containing untested paths with potential for untrapped errors."
>
> Replace it with the text "If the VSTL determines that additional module test cases are needed to provide coverage of all modules containing untested paths with potential for untrapped errors, the system shall be rejected and returned to the Manufacturer so they may develop the required module test cases and complete the subsequent internal system testing to bring the system back to the state of adequacy for submission for certification."

Explicit statement of Manufacturers module test coverage requirement

Recommendation: Explicit statement of Manufacturers module test coverage requirement Volume II

Appendix A Section A.3.3, Software Module Test Case Design and Data, second paragraph states "If the manufacturer's module test case design does not provide conclusive coverage of all program paths, then

the VSTL shall perform an independent analysis to assess the frequency and consequence of error of the untested paths."

We suggest that as this is the only location in the document that this requirement, that the manufacturer's module test case design must provide conclusive test coverage of all program paths, is stated, that it should be explicitly made in another location specific to the Manufacturer requirements.

We recommend the following be include in Volume I, Section 8.6, Quality Conformance Inspections, as the third bullet point under the second paragraph "c. Validate that the module testing provides conclusive coverage of all program paths"

Path coverage is an impossible requirement

Recommendation: Path coverage is an impossible requirement

Volume II, Section A.3.3, Software Module Test Case Design and Data, states, that "The VSTL shall **evaluate the test cases for each module**, with respect to **flow control parameters and data on both entry and exit**." Additionally it continues in the next paragraph to ask "If the manufacturer's module test case design **does not provide conclusive coverage of all program paths**, then the VSTL shall perform an independent analysis to assess the **frequency and consequence of error of the untested paths**. The VSTL shall design additional module test cases, as required, to **provide coverage of all modules containing untested paths with potential for untrapped errors**."

Path coverage, unless qualified, demands that every possible route through a module is executed. It has been proven that full path coverage is usually impractical or impossible. See Wikipedia article on Code Coverage for a more thorough discussion of the issue.

We suggest the following changes be made to the standard, that the first two paragraphs of Section A.3.3 be replaced by the following, and that the third paragraph remains unchanged:

> The VSTL shall review the manufacturer's program analysis, documentation, and module test case design. The VSTL shall perform an independent analysis of the test cases and test results for each module to determine the statement coverage provided by the module test cases, and the consequence of errors within any untested statements. This assessment will include a rigorous evaluation of the flow control parameters and data at both entry and exit points of each module.

If the manufacturer's module test case design does not provide conclusive coverage of 100% of the module's statements or problems are discovered in the evaluation of module flow control or entrance/exit points, the VSTL shall design additional module test cases, as required, to provide coverage of all modules containing untested statements and resolve the gaps identified in flow control, and entrance/exit processing. All discrepancies between the Software Specifications and the test case design proposed by the VSTL shall be corrected by the manufacturer prior to initiation of certification testing.

Volume II Appendix B (2005 V1.1) National Certification Test Report

Recommendation: Rename Witness Build to official EAC title of Trusted Build

We suggest the following changes to Volume II, Appendix B: National Certification Test Report, to bring it in alignment with the current EAC regulations.

Under Section B.1, Test Report Format, Subsection 3, Test Findings and Recommendation, change "Appendix C. Witness Build" to "Appendix C. Trusted Build".

Additionally, under Section B.2, Required Content of Test Report, change Subsection heading "Witness Build" to "Trusted Build". For the subheading text, make the following change from "The VSTL shall include as Appendix C of the Test Report a copy of the record of the final witness build and sufficient description of the build process to enable reproduction of the build." to "The VSTL SHALL include as Appendix C of the Test Report a copy of the record of the trusted build, as defined in the current version of the EAC's *Voting System Testing & Certification Program Manual*, to provide sufficient description of the build process to enable reproduction of the build."