Test Assertions
What are they and why do we need them?

Mark Skall

My Background

- Past Division Chief of Software Division at NIST
- Led Voting Project
- Retired from NIST in 2009
- Remained active in Voting
  - EAC
  - NIST

Testing Requires Unambiguous Requirements

- Need mutual understanding of VVSG requirement among voting system manufacturers, VSTLs, NIST and the EAC
- The “devil is in the details” to unambiguously specify requirements
- Test assertions can provide that mutual understanding among the EAC, NIST, manufacturers and VSTLs

What are Test Assertions?

- Conditions that must be met to determine conformance to specific requirements in the VVSG
- Each requirement is broken down into specific, unambiguous, testable conditions
- One or more test assertions for each requirement

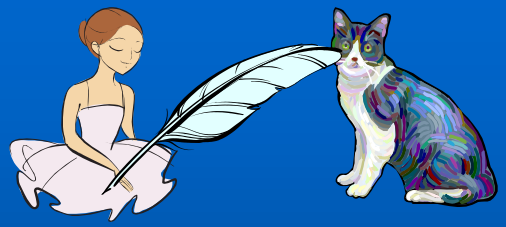
Why are they important? For current VVSG

- Currently each VSTL develops their own set of test cases to test VVSG requirements
- Since test cases are proprietary there is no way for public to scrutinize them for completeness or correctness
- Different test cases lead to different ways to test – no consistency across VSTLS
- Can result in different pass/fail results
- VVSG requirements can be high-level, vague, open to interpetation and ambiguous

English is not Precise

English is not Precise

- The girl touched the cat with a feather
  - (Girl + feather) touched cat



  - Girl touched (cat + feather)

Interpretation Issues

- Permit the voter to cast a ballot *expeditiously*
- Function *properly*
- Does not introduce any *bias*
- Provide *clear* instructions
- *Consistent* relationship
- *Maximize* correct perception
- *Minimize* cognitive difficulties
- Presented in an *equivalent* manner

Two possibilities for each requirement

- Precise and clear
  - TAs break it down into testable components
- High-Level, vague or ambiguous
  - Achieve consensus on meaning and interpret through test assertions
    - Can occasionally be subjective
    - Same subjective interpretation shared by all VSTLs

Example of a Test Assertion

- VVSG Requirement – Each module shall be mnemonically named
  - Test Assertion: IF a class, interface or callable unit is declared, THEN its intrinsic purpose can be determined by its name.

Examples

- **VVSG 1.0 Requirement 3.1.6a:** Voting machines with electronic image displays shall not require page scrolling by the voter.
- Assertions:
  - **TA316a-1:** IF a voting machine contains an electronic display THEN there SHALL be no off-screen contents that can be made visible solely through the use of scroll bars.
  - **TA316a-2:** There SHALL exist at least one mechanism, other than scrolling, for navigation within and between contests that presents ALL ballot-content to the voter explicitly.
  - **TA316a-3:** Next or previous “page” buttons MAY be used as such a non-scrolling navigation mechanism.

Examples

- **VVSG 1.0 Requirement 3.1.4a:** In both visual and aural formats, contest choices shall be presented in an equivalent manner.
- Assertions:
  - **TA314a-1:** FOR all contest choices on a visual ballot, there SHALL be no discernible differences in visual presentation.
  - Font properties (bold, italic, underline)
  - Text properties (word and letter spacing, etc.)
  - Visual presentation of color
  - Many more . . .

Assertion Project

- An effort to provide a reference set of assertions that are complete, unambiguous, and:
  - *Provide a uniform testing reference* for VSTLs and voting system manufacturers, across all testing domains (security, usability, software requirements, performance, etc.)
  - *Provide a "bridge"* between the VVSG requirements and test suites (manufacturer's, VSTL's or NIST's)
  - *Provide testable expressions* (assertions) that more succinctly and practically describe adherence to normative VVSG requirement statements.

Team Effort

- This is a team effort among NIST, EAC and VSTLs
  - Everyone has to agree before test assertion is finalized
  - Made available to manufacturers for their comments
  - Decisions are somewhat subjective but better to interpret these one time by a consensus than having VSTLs interpret them unilaterally and inconsistently

Process

- Team consists of myself plus NIST and EAC
  - Domain Experts
- I develop draft assertions for requirements
- Team meets and discusses, modifies, etc.
  - Team achieves consensus
- Distribute to VSTLs for feedback
  - Review VSTL feedback and modify
- Distribute to manufacturers
  - Review manufacturer feedback and modify
- Post final assertions on NIST web site

Status

- Test Assertions completed for VVSG 1.0
  - Usability and Accessibility(Section 3)
  - Security (Section 7)
  - Software (Section 5)
    - Done previously
    - Different process
    - Different syntax
- Test Assertions for VVSG 1.1
  - QA/CM (Section 8)
  - Security (Section 7)
  - Usability and Accessibility (Section 3)

Future Plans

- Test Assertions for rest of VVSG 1.1
- Goal is complete set of assertions for entire standard
- Compete set distributed (and mandated) for use by VSTLs

Benefits

- Ensures that each requirement is tested correctly and comprehensively
- Helps ensure that testing is uniform and consistent among all VSTLs
  - Ensuring same pass/fail result regardless of which Laboratory is used
- Clarifies high-level or vague terminology
- Manufacturers can determine what is expected for each requirement

Implications for New VVSG

- Lessons learned in developing and specifying requirements
  - Make sure all terms/words are clear and understood by all
  - Think of possible test assertions
- Lowest level of the new VVSG
- Testable Level
- Formal specs?