Software Validation and Verification for Non-software Engineers
(DIY Software Validation)

Bob Kuczwarskyj
October 14,2014

Agenda/Topics

- Why validate
- Validation, Verification, Qualification, Testing?
- Scope - Off the Shelf Software
- Validation goal is confidence in computerized system

Agenda/Topics – Cont.

- How to do it – start with end in mind
- Determine intended uses
- Document requirements
- Risk assessment/management
- Create Validation Plan
- Document Testing Scripts
- Execute Testing Scripts
- Create Report
- V&V Lite

Why Validate

- Proof of meeting standards
- Required by regulations
- Safety of end user, operators, customers
- Good business sense
  - Meet user needs (i.e., Apple)
  - Identify defects early – 100 to 1 benefit ratio if found in requirements phase
  - Reduce recalls
- "I don't trust software"

Software is subject to systemic risk (subject to failure due to design, specification, operating, maintenance, and installation errors – risks inherent in the system and methods).

Validation, Verification, Qualification, Testing?

- Myth – Valid. = Verif. = Qual. = Testing
- Definitions?
  - Validation – Built the right product (meet user needs)
  - Verification – Built the product right (followed process)

Scope – Off-the-Shelf Software

- Rough categorization of software into custom built and off-the-shelf (OTS)
- Custom built SW is validated by the development organization – software engineers using a methodology
- OTS = out-of-the-box, open source, configurable
- Level of control constrains validation activities (design and “coding” phases inaccessible)

Validation Approach

- An approach:

Identify the most meaningful set of confidence building, value-added activities to be applied during the software life cycle.

(based on AAMI TIR 36:2007)

- Use critical thinking to analyze various aspects and risks of the software and select items from a toolbox to build conclusion that software is validated.

Start with end in mind

- Configuration management/change control of documents and software (all artifacts)
- Defect reporting – visibility of status of “bugs”/anomalies
- Documentation – e.g., user manuals, plans, reports, “objective evidence”
- Standard Operating Procedures – Work Instructions
- Training

Intended Uses – 5 “W”s

- Explore intended use to focus on purpose of the software – precursor to requirements
- Where – International, office/machine shop, mobile/embedded device, Hazardous/clean room
- Who – Researcher/salesman, dashboard viewer, data entry, data acquisition device
- What – CRUD – create, report, update, delete (archive), disaggregate, calculate

Intended Uses – 5 “W”s – Cont.

- When – 24/7, event triggered, cycle, ad hoc, consideration of future modification
- Why – reasons for using the software
  regulatory, financial, efficiency,
  competitive pressures

Intended uses define what is in and out of scope.

Document the Requirements

- Inputs
- Outputs
- Interfaces
- Error messages
- Data Definitions
- Processes – Procedural steps
- Usability
- Characteristic – Traceability and Testability

Document the Requirements – Cont.

- Gathering requirements
  - Users of system
  - Documents – forms, reports
  - Pre-existing system
  - Technical Standards
  - Interviews
  - Focus groups
  - Prototypes

Document the Requirements – Cont.

- Review and vet requirements
- Document with flow charts, decision tables checklists – make as understandable as possible
- Include system requirements – security, backup, environment, documentation, user manuals, training, safety, hardware, performance
- Spend 30% of project time on requirements – 100 to 1 pay back in reducing defects

Risk Assessment

- Conduct after requirements developed
- Software – probability can not be determined, focus on severity
- Feedback the mitigation actions into the requirements

Create Plan
Sections Include:

- Scope
- Background
- System Description – benefits of project
- References – standards, regulations
- Responsibilities
- Validation Approach – constraints due to level of control
- Assumptions

Create Plan – Sections –Cont.

- Risk Assessment
- Requirements document
- Vendor Qualification – how do they validate their software
- Validation steps/execution
- Configuration Management (consider entire life cycle of software)
- Training Plan
- Validation deliverables

Create Test Scripts

- Form with: instructions/steps, expected results, space of actual results, Pass/Fail, and signature columns
- Requirements document is basis of tests
- Installation Qualification – IQ
  - Hardware platform – desktop, laptop, mobile, web
  - Operating systems – versions of Windows, Linux
  - Environment

Create Test Scripts – cont.

- Operational Qualification – OQ
  - Initial operation
  - Key functions
  - Nominal testing
  - Functional Parameters
- Performance Qualification – PQ – for software, it is end-user testing in the actual environment

Testing Techniques

- Normal Path
- Boundary value testing
- Cross check of calculations
- Cross check of audit trail
- Negative testing - Error checking
- Output testing
- Exploratory/Ad Hoc – record steps

Execute Activities

- Use project plan or Gantt chart to track activity – develop a source for metrics
- Record anomalies and their resolution
- Document execution – no evidence means you can’t prove it happened

Create Report

- Description of tested system
- Project activities
- Documentation of complete test protocols
- List of anomalies and resolution
- Final Approval or rejection
- Results of Review

V&V Lite

- Create a Plan
- Requirements – make them testable and traceable
- Risk Management
- Test scenarios and detailed test scripts
- Execute and document execution
- Report - detailing environment, hardware, software, results, anomalies, conclusion

Summary

- Objective is building confidence in software
- Elements are intended uses and risks and the resultant requirements ….
- Choose tools for validation activities
- Document and review – provide the objective evidence to support plan and protocol execution

References

- Association for the Advancement of Medical Instrumentation - Technical Information Report TIR36:2007 “Validation of software for regulated processes”
- Wiegers, Karl, “Software Requirements”, Microsoft Press, 2003
- Vogel, David, “Medical Device Software – Verification, Validation and Compliance”, Artech House, 2011
- Beizer, Boris, “Software Testing Techniques”, International Thomson Computer Press, 1990