Software Verification and Validation

AMDM-April 20, 2012

Andrew Grove, PhD
Reviewer, Division of Microbiology Devices
OIVD/CDRH/FDA/HHS

andrew.grove@fda.hhs.gov

U.S. Department of Health and Human Services
Food and Drug Administration

Validation Guidance

General Principles of Software Validation; Final Guidance for Industry and FDA Staff (2002)

(http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM085371.pdf)

Regulatory Requirements

- Software Validation is a requirement of the Quality System regulation.
- Validation requirements apply to:
  - software used as components in medical devices.
  - software that is itself a medical device.
  - software used in production of the device or in implementation of the device manufacturer's quality system.

Some Definitions

- **Requirement**-can be any need or expectation for a system or for its software.
- **Software Requirements**
  - Typically derived from the system requirements for those aspects of the system functionality that have been allocated to software.
  - Stated in functional terms and are defined, refined, and updated during development.

- Success in accurately and completely documenting software requirements is a crucial factor in successful validation of the resulting software.

Some Definitions Cont.

- Software Verification- provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase.

Some Definitions Cont.

- Software Validation- confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.

Software Validation

- Critical tool used to assure the quality of device software.
- Can increase the usability and reliability of the device resulting in decreased failure rates, fewer recalls and corrective actions, less risk to patients and users, and reduced liability to device manufacturers.
- Can reduce long term costs by making it easier and less costly to reliably modify software and revalidate software changes.

Principles of Software Validation

- Requirements
  - Established software requirements
- Defect Prevention
  - Software testing is necessary but, by itself, not sufficient to establish confidence. Need a mixture of methods and techniques.
- Time and Effort
  - Should be based on evidence collected from planned efforts conducted throughout the software lifecycle.

Principles of Software Validation-Cont.

- Software Lifecycle
  - Takes place within the environment of a well established lifecycle. (contains software engineering tasks and documentation necessary to support the effort)
  - FDA does not recommend any particular lifecycle models.
- Validation Plans
  - Defines “what” is to be accomplished through the validation effort. (scope, approach, resources, schedules, types and extent of activities, tasks and work items)

Principles of Software Validation-Cont.

- Procedures
  - Establish “how” to conduct the validation effort. (specific actions or sequence of actions to complete individual activities, tasks, and work items)
- Validation After a Change
  - Analysis should be conducted not just for validation of the individual change, but also to determine the extent and impact of that change on the entire system. An appropriate level of regression testing should then be conducted.

Principles of Software Validation-Cont.

- Validation Coverage
  - Should be based on the software's complexity and safety risk associated with the use of the software for the specified intended use.
- Independence of Review
  - When possible, an independent evaluation is always better. (3rd party or internal staff members not involved in a particular design or its implementation)
- Flexibility and Responsibility
  - Manufacturer has flexibility in choosing how to apply these principles, but retains ultimate responsibility for demonstrating validation.

Verification, Validation, and Testing

- Difficult because developer can't test forever, and it's hard to know how much evidence is enough.
- Is a matter of developing a "level of confidence" that device meets all requirements and user expectations.
- Defects found in specifications documents, estimates of defects remaining, testing coverage, other techniques are all used to develop an acceptable level of confidence.
- Level of confidence, and therefore the level of software V&V and testing needed, will vary depending on the safety risk (hazard) posed by the automated functions of the device.

Software Testing Principles

- The expected test outcome is predefined.
- A good test case has a high probability of exposing an error.
- A successful test is one that finds an error.
- Testers use different tools from coders.
- Examining only the usual case is insufficient.
- Test documentation permits its reuse and an independent confirmation of the pass/fail status of a test outcome.

Software Testing

Should include:

- Unit level testing
- Integration level testing
- System level testing

These tests should provide a thorough and rigorous examination of compliance with functional, performance, and interface definitions and requirements.

System Level Testing

Should address:

- Performance issues (e.g., response times)
- Responses to stress conditions
- 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

Types of Software Testing

- **White-box (code-based, structural)-** based on knowledge obtained from the source code, detailed design specs, and other development documents.
- **Black-box (definition-based, functional)-** based on the definition of what the software product is intended to do.
- **Testing of software changes-** should demonstrate change was implemented correctly and that the change did not adversely impact other parts of the software product (regression testing).

User Site Testing

- Any testing that takes place outside of the developer's controlled environment.
- Should take place at a user's site with the actual hardware and software that will be part of the installed system configuration.
- Accomplished through either actual or simulated use of the software being tested within the context in which it is intended to function.

User Site Testing

- Should follow a pre-defined written plan with a formal summary of testing and a record of formal acceptance.
- Documented evidence of all testing procedures, test input data, and test results should be retained.
- Some of the developer evaluations should be repeated including tests for a high volume of data, heavy loads or stresses, security, fault testing, error messages, and implementation of safety requirements.

User Site Testing

- Validation of use should involve testing with users that involves realistic use, objective performance information and subjective evaluation by the users.
- Part of the validation should address whether the system (alarms, displays, messages, feedback, etc.) enables the user to use the device correctly.
- Faults detected should follow the same procedures and controls as for any other software change.

Software Changes

- Regression analysis and testing
- Validation of specific change
- Level of effort needed is dependent upon the degree to which validation of the original software was documented and archived.
- Test documentation, test cases, and results of previous V&V testing need to be archived if they are to be available for performing subsequent regression testing.

VV&T Documentation

Moderate Level of Concern:

- Describe VV&T activities at the unit, integration and system level.
- Include error code, and result flag validation, system level test protocols, pass/fail criteria.
- Testing results (linked to the Hazard Analysis and software functional requirements).

VV&T Documentation

- Specific process and elements to be considered include a discussion of testing results and a discussion of how the following was tested:
  - fault, alarm, and hazard testing
  - error, range checking, and boundary value testing
  - timing analysis and testing
  - special algorithms and interpretation tests and analysis
  - stress testing

VV&T Documentation

- Specific process and elements (continued…)
  - device options, accessories, and configurations testing
  - communications testing
  - memory utilization testing
  - qualification of OTS software
  - acceptance and user site testing
  - regression testing

Common Issues-VV&T Documentation

- System level test protocols not included or not clearly identified.
- Pass/Fail criteria for each of the System level test protocols not included.
- Including a general pass/fail criteria for all system testing.
- Off-the-shelf software validation not included.

Other Guidance Documents

- Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices (2005) (http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089593.pdf)
- Off-The-Shelf Software Use in Medical Devices (1999) (http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/UCM073779.pdf)
- Guidance for Industry - Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software (2005) (http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm077823.pdf)
- Draft Guidance for Industry and FDA Staff - Radio-Frequency Wireless Technology in Medical Devices (2007) (http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm077272.pdf)

Questions?