When Medical Device Software Fails Due to Improper Verification & Validation (V&V)

Ephrat Most, ScD  
Senior System Engineer/QA

Nick Oshman, BsE  
Project Manager, Engineering

Bruce Swope, BS  
Vice President of Engineering
Software Failure

- Improper software design
  - Games / Coke machine
  - Medical Device
Software Failure and FDA recall

• Software may be
  – in a medical device,
  – may be itself a medical device, or
  – it may be used to manufacture a medical device
  – software added to a smart device may transform the device into a medical device.

• Medical devices increasingly rely on software and seemingly minor changes to software can have important implications for device function and clinical performance.

• Failure to implement software design controls, and develop sufficient testing procedures can lead to software anomalies often requiring a correction or removal.
Is a Software Considered Medical Device?

• Software used to plan cancer treatment doses and to control the setting of oncology treatment devices
• Software used to measure and calculate the anatomical bone landmarks for surgical intervention
• Software embedded in an implanted insulin measuring device
• Software used in the design and manufacturing to produce a medical device (e.g. customized implant)
• Software used to collect patient data for research purposes
Software Recall

• The FDA's analysis of 3140 medical device recalls conducted between 1992 and 1998 reveals that 242 of them (7.7%) are attributable to software failures.
• Of those software related recalls, 192 (or 79%) were caused by software defects that were introduced when changes were made to the software after its initial production and distribution.

General Principles of Software Validation, FDA, January 11, 2002, Section 3.1.2
Software Recall - CDRH

• CDRH conducts detailed analyses of recall data to better understand identified trends
• Common Cause of Recall – Software Design
• Number of software Case Recall events FY2008-2012 average of 15% of all CDRH recalls.
  – For example, in 2012: total of 746 Recalls
    • 12 Software change control
    • 169 Software design
    • 5 Software manufacturing process

Medical Device Recall Report FY2003 to FY2012, FDA Center for Devices and Radiological Health Office of Compliance Division of Analysis and Program Operations
Verification / Validation Definitions

• Verification
  – Should occur at every stage of the software development life cycle
  – Should demonstrate that all software requirements have been implemented correctly and completely and are traceable to system requirements

• 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.”

General Principles of Software Validation, FDA, January 11, 2002, Section 3.1.2
Verification vs. Validation

• Verification
  – Are we building the product right?
  – Are we developing the product/system/software correctly?
    • Need to look internally to verify that all specifications are implemented correctly and
    • That inputs and outputs comply with previously determined regulations or standard.

• Validation
  – Are we building the right product?
  – Are we developing the right product/system/software?
  – The act of ensuring compliance against an original requirement
    • Need to examine the specifications from the user’s point of view and
    • Ensure that the product meets the needs of the end user.
Verification Testing

- Code Review
- Unit Testing
- Integration Testing
- System Testing
Verification Testing

• Code review
  – Systematic examination of software source code

  – Benefits:
    • Finding bugs early.
    • Coding standards compliance
    • Consistent design and implementation
    • Higher software security.

  – Downside:
    • Majority of bugs that are caught in code review are trivial bugs. The bugs that actually take time to find don't get caught in review.
Verification Testing

• Unit Test
  – A level of software testing where individual units/ components of a software are tested
  – To make unit testing possible, the codes needs to be modular. This means that codes are easier to reuse

  – Benefits:
    • Increases confidence in changing/ maintaining code.
    • Codes are more reliable
    • The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
    • Development is faster

  – Downside:
    • There is of course the up front cost of writing the unit tests
    • Unit tests a small part of the entire software
Verification Testing

• Integration test
  – When individual software modules are merged and tested as a group → integration testing
  – Aims to reduce the need for scaffolding code by using the actual code modules as they are developed.
  – It verifies the proper execution of software components and proper interfacing between components within the solution
Verification Testing

• System test
  – The behavior of whole system/product is tested as defined by the scope of the development project or product
  – It may include tests based on risks and/or requirement specifications, business process, use cases, or other high level descriptions of system behavior, interactions with the operating systems, and system resources.
  – System testing is most often the final test to verify that the system to be delivered meets the specification.
  – System testing is carried out by specialist testers or independent testers.
  – System testing should investigate both functional and performance requirements of the testing.
Validation Test

• Internal / External study
• Usability / User acceptance testing
• Acceptance Test
• Gold standards

• Advantages:
  – Accompaniment the verification
  – Provides a better understanding of the difference between the actual result and expected result
  – Validation helps in building the right product as per the customer’s requirement and helps in satisfying their needs.
Case Study: Image Processing Software

- Image Capture
- GUI (View, image manipulation)
- Image Analysis / Cell Identification
- Data Analysis
Did We Get the Results We Wanted?

• Verification Aspect
  – Code
    • Conform to standards
    • Written properly and make sense
    • Error free (code review)
  – Unit testing
  – Integration testing
  – System (black box) testing

If all verification tests passed BUT we marked the wrong cells – we have an issue
Did We Get the Results We Wanted?

• Validation Aspect
  – Yes we identified the correct cells
    • Gold Standard (hemocytometer)
    • User accepted the cells identified
  – But
    • Is the code error free?
    • Will the code produce a different output for a different system setting?

If all validation tests passed BUT we our code has not been tested – we also have an issue
Proper V&V

- Verification + Validation
  - All tests passed

- Gold standard Validation:
  - Yes we have accurately counted the identified cells and compared to a gold standard (e.g. hemocytometer)
Validation of Off The Shelf Software

• OTS presumes that the software in question is a purchased product that is designed, developed, and supported by other company
  – Only part of its functionality apply to your

• All third party tools (and libraries) used to construct or test FDA regulated software need to be validated

• Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices, September 9, 1999
Conclusion

• Verification and validation are critical components in the development life cycle of any software.
• Verification starts from the requirements analysis stage where design reviews and checklists are used.
• Validation is meeting user needs and intended use, which will contain some functionality.
• The results of the V & V process are imperative to the safety of the medical device.
• One of the biggest issues: how much V&V is enough?
  – More testing → potential better results
  – Time and cost are also affected
  – Failure is liability → liability is money
  – SW FDA Classifications
Acknowledgment

• Sterling Medical Devices
  – Dan Sterling
  – Bruce Swope
  – Nick Oshman
When Medical Device Software Fails Due to Improper Verification & Validation (V&V)

Ephrat Most, ScD
Senior System Engineer/QA

Nick Oshman, BsE
Project Manager, Engineering

Bruce Swope, BS
Vice President of Engineering