MDTMAG.COM | MARCH 2017
MDT MEDICAL DESIGN TECHNOLOGY
The Medical Design Engineer's Resource for Products & Technologies

Testing Error Conditions in Control Software for a Heart Pump

By Walt Maclay, Vder Systems

Medical devices are incorporating a significant amount of software to achieve higher levels of functionality, more robust and reliable operations, and greater resiliency by detecting and managing a complex set of error conditions. This increase in software content has resulted in some widely publicized device failures, including some deaths. As a result, the U.S. Food and Drug Administration (FDA) has implemented extra requirements on software. In this article, we will share the software techniques that we used for the verification for a Class III heart pump. Since the functional requirements for this instrument include hundreds of different alarm conditions, a large portion of its software is dedicated to dealing with error conditions and alarm reports. All of them had to be thoroughly documented and verified for FDA approval.

Software Complexity

Device safeguards and error handling accounted for 50 to 80 percent of the actual software code and about 70 percent of the software development challenges. The basic code we created to manage the input and output measurements and to control the speed of the pump's motor was relatively simple. However, the additional functions needed to insure that it kept on running under nearly any conceivable error condition (including power cord disconnection) added significantly to the complexity of the final design. Understanding this source of complexity allows you to be more predictable in managing software projects and lowers the risk with careful planning and management of the software complexity.

Built-in Diagnostics

Device safeguards include several types of diagnostics:

- Startup Diagnostics - These automatic self-tests are performed when the device is first turned on
- Run-Time Diagnostics - These real-time tests are performed to diagnose faults during the operation of the device
- Extended Diagnostics - These tests contain a suite of diagnostics that will test the device under user control

Software-Related Documentation and Verification

The best practice approach for verification is to systematically look at the safety and system risk issues. We have several broad categories of error conditions:

- Use Errors - Both for unintended and intended use, user errors, and environment use errors
- Output Errors - Graphical User Interface (GUI) text, help information, lights, alerts, and alarms sounds
- Workflow Errors – Flowcharts and diagrams for user guidance, alerts, alarms, GUI, and help information
- Performance Errors - Flow rate control and speed of motor control
- Processing Errors - Incorrect handling of data or data calculation errors
- Input Errors - Incorrect patient measurement and conditions including corner cases
- Hardware Failure - Processor failing or a broken or disconnected cable

Broad Categories of Error Conditions
Use Errors
Output Errors
Workflow Errors
Performance Errors
Processing Errors
Input Errors
Hardware Failures

Figure 1. A list of error condition categories

Test / Verification Environment

Verification involves the execution of the heart pump to ensure that it meets the requirements and responds correctly in a method that is in accordance with the FDA guidelines. Testing ensures it performs its functions.

We created a test environment around the device that simulates the environment under a variety of both real and potential error conditions. It allowed us to verify the device under all kinds of inputs and loads. Our test

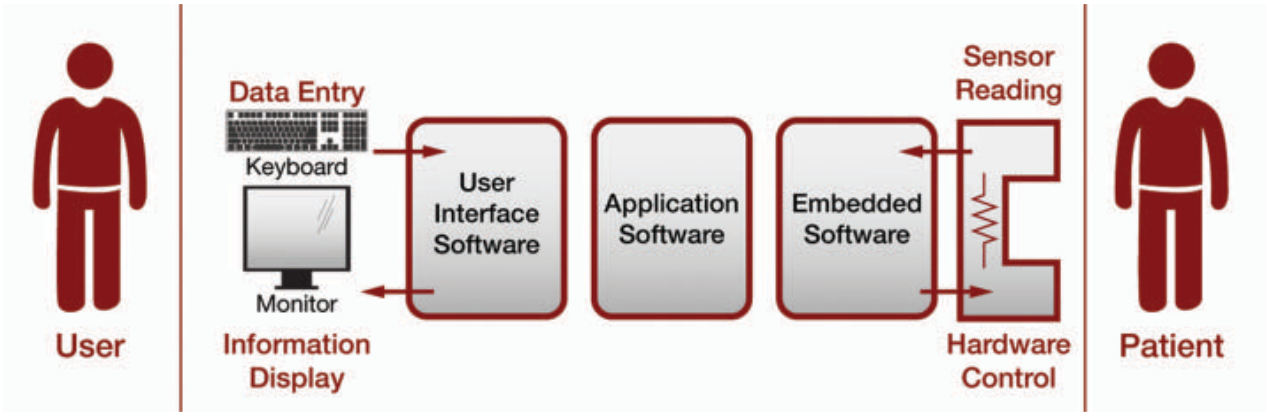

Figure 2. Software segmentation within a typical medical device

environment allowed for fault injection. Fault injection is a technique where the effectiveness of tests and diagnostics can be assessed by creating simulated faults and seeing their effect on the system. We also used programmable signal generators to simulate abnormal sensor inputs and conditions.

Managing Risk / Risk Analysis

It is important to understand the difference between verification and validation. Verification tests that the product meets the specifications/ requirements as written. Verification answers the questions:

- Did we build the product as we defined it?
- Is it bug-free?

Validation ensures that the product meets a real need and that the right product is built. This can only be done in a clinical setting when you have a heart pump actually pumping blood through a patient.

Software verification and validation is only one portion of the overall equation. To increase confidence in safety and reliability measures, all aspects of the software development life cycle must be monitored for compliance with applicable requirements, standards, procedures, and regulations. This can only be accomplished through rigorous quality assurance activities.

Risk analysis and providing traceability must be performed at all stages of the development process: user needs, detailed functional requirements, architecture, design, reviews, verification, test cases, defect reporting, contingency, and next step planning. This management is a mandated requirement, whether a traditional waterfall or more iterative approach, and is used for the software development. Impact analysis and version history must be complete.

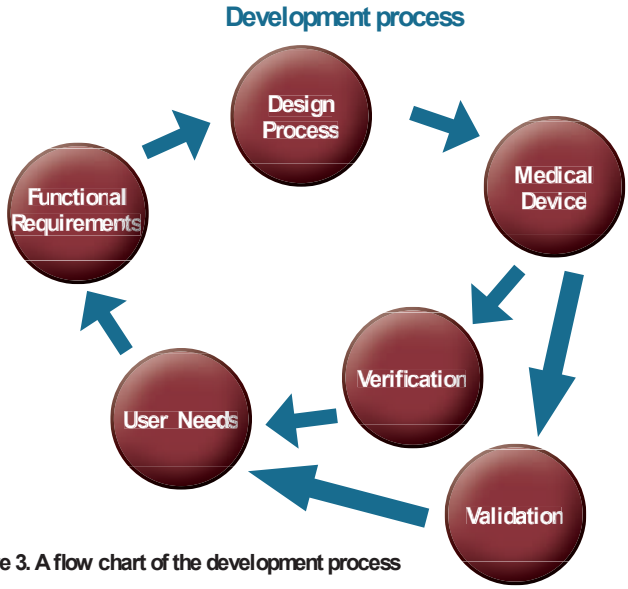

Figure 3. A flow chart of the development process

FDA Submission and Approval

Our client submitted to the FDA for 510(k) clearance and we are pleased to say that it was approved without any conditions or comments. It was quite unusual that the FDA had no comments on software for a Class III device. Normally when you do an FDA submission on any device that has software, most of the issues are around software.

In conclusion, software is a major risk for medical devices and can complicate getting FDA approval. A systematic approach to verification and traceability analysis can help you prove to the FDA that you have tested your system against regulatory standards and it meets clinical needs without unnecessary patient safety risk. MDT