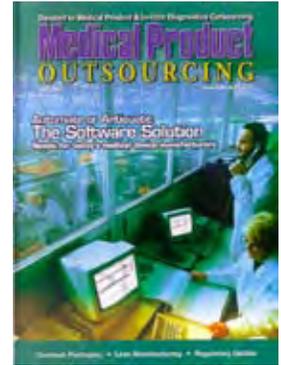# Solid Strategy, Sound Management:
Keys to Testing Software

by David A. Vogel
Intertech Engineering Associates, Inc.

as published in <u>Medical Product Outsourcing,</u>
April, 2006

> *Testing is but one activity in a good software validation program. A good testing program encompasses a diverse collection of test protocols. However, when taken seriously testing (along with all validation activities) can produce solid software that can stand the test of time in the field.*

System-level testing, also known as "black-box testing," tests software with no knowledge of the design and implementation details of the software. On the other hand, unit-level (white-box) testing individually tests each unit for function, subroutine, and other software standards and is based on intimate knowledge of design and implementation details. Integration-level testing lives somewhere in the middle and is mostly concerned with the interactions and interfaces between modules of code.

With all three types, well-written test procedures must detail every step the tester should take. Highly detailed procedures increase the probability that others will be able to re-create any defect observed and assure consistency in testing results from tester to tester. A high level of detail also makes it easier to bring relatively non-expert testers into the project. Not only does this add resources to a testing program, it also has another benefit: inexperienced testers often see obvious defects that the experienced testers have become blinded to and accept as normal.

Ad hoc tests go a step further in this direction. They are intentionally not detailed to introduce an element of randomness into the test process. *Ad hoc* tests generally ask the tester to accomplish a task without providing any details about how to do it. This sets up the tester to accomplish the task in ways that would differ from the detailed procedures. User error can exercise parts of the code that may not be normally exercised because the developer didn't anticipate the user errors.

COMPANY PROFILE

Intertech Engineering Associates, Inc.

Address: 100 Lowder Brook Avenue

   Suite 2500
   Westwood, MA 02090

   www.inea.com - (781) 801-1100

Industry: (Electro)Medical Devices

Services: Assessments
   Training
   Consulting
   Hands-on Engineering

Skills: Product Design
   Risk Management
   Requirements Engineering
   Electronics Development
   Software Development
   Software Verification and Validation
   Production/Quality System Software Validation

Surprising Considerations for Useful Testing

A good testing program encompasses a collection of test protocols of diverse types. Good test designs, test execution and documentation are all necessary for useful test outcomes. Several soft factors also strongly influence the success of a testing program.

A tester's experience level. The experience level of the tester and test designer is highly correlated with success in testing (finding defects). Experienced testers are well trained in the skills of observation and seldom intimidated from asking questions or writing up observations. They have a sixth sense about where they will find defects in the software. Much of this has to do with past experience in other testing programs or in their past lives as developers. Inexperienced testers, however, tend to make the unexpected mistakes that often uncover defects the experienced tester may never encounter. A diverse collection of testers with different experience levels yields best results.

Software complexity. The complexity of the software is also related to the likelihood of finding defects. However, test designers too often under-test complex functionality because they may not understand it well. Paradoxically, the most complex - and most likely to be defective - software is tested least. Managers need to be aware of this tendency and work diligently to focus efforts in the complex areas most likely to yield defects.

Independent (objective) evaluation. Independence is another soft attribute that affects test outcomes. A developer cannot be objective enough to produce reliable test results on his or her own code. However, other types of independence also can affect outcomes. It's desirable to have those who execute tests be independent from those who design them. This will help surface errors in the test procedures themselves. It is even desirable to have independent groups of testers rotate in and out of a testing program to reduce tester fatigue. Tester fatigue allows testers to become too familiar with the software and, thus, overlook anomalous behaviors they have come to accept as normal.

Even independence of organizational responsibility is important. Testers and test designers should not report to the management that is responsible for development of the software. A tester's success in finding a defect is likely to be viewed as a development failure (delaying release dates, etc.). This obvious conflict of interest will impact the quality of test outcomes if the tester is blamed for delaying the project.

State of Mind. This is also important to a successful testing or validation program. All team members, developers, designers and testers need to be coached that finding defects, reporting them and tracking them to resolution represents success. These are defects that will not make it to the field and potentially become life-threatening.

Can You Over-test?

The flip side of under-testing complex software is over-testing simple functionality. Testers have a tendency to "camp out" in areas of functionality in which they are comfortable. Management should be aware of this tendency and look for signs of it in schedules, plans and in the test procedures themselves.

Test engineering management should constantly channel resources to where they will produce the most benefit and find the most defects; that usually is in the most safety critical code or the most complex code. Over-testing simple functionality or well-exercised functionality is not a good investment of scarce testing resources.

Sometimes, certain functionality exists in many places in software-driven systems. For example, a data-input field may appear identically in each of a number of operating modes for the device. It may not be necessary to do full testing of that input field in every instance (eg, normal/abnormal inputs, boundary values, etc.) if one can assume that the software implementing that field is common for each mode in which it is used. In cases like this, the test designer can identify an equivalence class of functionality that is tested once and assumed to be applicable to all occurrences in the system. This greatly reduces the testing.

Exercise care when using equivalence-class logic. The reduction in test effort is only valid if the assumption

"Solid Strategy, Sound Management: Keys to Testing Software"
as published in: <u>Medical Product Outsourcing</u> - April, 2006

page 2
Copyright 2006 - Intertech Engineering Associates, Inc.

behind the equivalence relationship is true. Every assumption should be documented and tested to assure that the equivalence-class logic is valid.

How do you know how much testing is enough testing, and when to stop? Ideal defect-testing finds defects at a steady rate until testing is complete, and no defects are found after that. More typically, however, the defect-testing function is asymptotic. One never knows for sure that all defects have been identified and repaired.

A test plan - written and agreed before the testing process begins - should contain some thought about what the strategy will be to end testing. Things to consider in devising the "stop test" policy are:

- Number of unresolved defects
- Number of reported defects that could not be reproduced
- Severity of defects reported on last test run
- Rate of defect identification

This policy will differ for all devices, depending on their criticality in the clinical setting, their size and their complexity.

Testing is but one activity in a good software validation program, and is not by itself sufficient to validate software, according to the FDA's definition. Testing is a bona fide discipline of its own and is part art, part science, part common sense and part good organization. When taken seriously, testing - and all validation activities - can produce solid software that can stand the test of time in the field.

"Solid Strategy, Sound Management: Keys to Testing Software"
as published in: <u>Medical Product Outsourcing</u> - April, 2006

page 3
Copyright 2006- Intertech Engineering Associates, Inc.

ABOUT THE AUTHOR:

David Vogel is the founder and president of Intertech Engineering Associates, Inc.

Dr. Vogel was a participant in a joint AAMI/FDA workgroup to develop a standard for Critical Device Software Validation which was subsequently included in the IEC 62304 Software Lifecycle Standard. He was also a participant on the joint AAMI/FDA workgroup to develop a Technical Information Report (TIR) for Medical Device Software Risk Management. Currently, Dr. Vogel is a member of the AAMI/FDA workgroup developing a TIR on Quality System Software Validation.

A frequent lecturer for workshops and seminars on topics related to medical device development and validation, Dr. Vogel also is the author of numerous publications and holds several patents.

Dr. Vogel received a bachelor's degree in electrical engineering from Massachusetts Institute of Technology. He earned a master's degree in biomedical engineering, a master's degree in electrical and computer engineering, and a doctorate in biomedical engineering from the University of Michigan.

Intertech Service Offerings:

Risk Analysis and Management
Software Design and Development
Electronic Design and Development
Requirements Development and Management
Documentation and Traceability
Verification and Validation
Evaluations, Reviews, Inspections
Planning
Project Management
Compliance Consulting and Training
Manufacturing and Quality System Software Validation

Leverage INTERTECH's expertise to:

Reduce Project Risk
Shorten Time to Market
Cut Development and Test Cost
Assure Quality Products

ABOUT INTERTECH:

Intertech Engineering Associates has been helping medical device manufacturers bring their products to market since 1982. Through a distinct top-down service model, Intertech offers high-level consulting and hands-on engineering. By balancing technical expertise and practical business experience, we support clients through all phases of product development. While we do make your job easier, Intertech exists not to replace but to partner with clients to help balance the concerns of quality, time and cost.

With considerable experience in FDA regulatory compliance, our time-tested development process can anticipate and solve problems inexpensively on the planning board rather than through costly solutions later in the development, test, or post-deployment phases. By using deliberate processes, Intertech ensures an improvement in quality and can build client expertise.

Call us today for more information or a free consultation at 781.801.1100

INTERTECH Engineering Associates, Inc.
100 Lowder Brook Drive Suite 2500 Westwood, MA 02090 USA

www.inea.com - info@inea.com - Tel: (781) 801-1100 - Fax: (781) 801-1108