PARASOFT®

WHITEPAPER

Dramatically Reduce Risk

Achieving Code Coverage Requirements in Class II & Class III Medical Devices

EXECUTIVE SUMMARY

In an increasingly software-intensive development process for medical devices and equipment, software can ease the regulatory burden, business monetary risk, and even the moral aspect of finding bugs earlier by having more comprehensive testing in the form of software code coverage tools. These so-called test coverage metrics are enshrined in U.S. FDA guidance and are part of the FDA's 510(k) process, though not all that clearly.

While the level of the various types of code coverage varies between 510(k) Class II and Class III devices, there's clear evidence that untested functionality increases risk. The FDA focuses heavily on risk analysis and risk mitigation.

Developer-friendly automated software testing solutions like Parasoft C/C++test and Parasoft DTP provide the ability to:

» Measure various types of code coverage automatically.

» Integrate with modern software development life cycle (SDLC) processes and software tools.

The result? Decreased risk and increased confidence in the safety and security of medical devices.

Automated software testing solutions for embedded systems help teams meet FDA recommendations and ease the burden of preparing documents for regulators. Interestingly, Europe is increasingly reliant upon the U.S. FDA, in part because the medical product discovery and development process has become increasingly globalized.

INTRODUCTION

Most of us, at one time or another, have been the beneficiaries of the features and functions of medical devices, also known as medical equipment. From the common blood pressure cuff to the inexpensive pulse oximeter, SpO2 meter, you can buy at your local pharmacy, from clinician operated CT scanners to MRI machines, all are subject in one way or another to U.S. government regulation, through the agency known as the U.S. Food and Drug Administration (FDA).

As the FDA puts it:

> Device classification depends on the intended use of the device, and upon indications for use. In addition, classification is risk based, that is, the risk the device poses to the patient and/or the user is a major factor in the class it is assigned. Class I includes devices with the lowest risk and Class III includes those with the greatest risk. As indicated above all classes of devices [are] subject to General Controls. General Controls are the baseline requirements of the Food, Drug and Cosmetic (FD&C) Act that apply to all medical devices, Class I, II, and III.al Equivalence in Premarket Notification [510(k)].[1]

Most medical devices regulated by the FDA require an FDA 510(k) premarket notification or a simple registration if exempt from 510(k) requirements. However, if the medical device sustains or supports life, if it's implanted or presents a "potential unreasonable risk of illness or injury," the medical device is likely a Class III device, which will require premarket approval (PMA) from the FDA before it can be marketed in the U.S. Class II devices, though, are much more common and are serious enough to fall under the purview of the FDA.

New kinds of medical devices that are so innovative that there are n't any previous examples or substantial equivalents (called "predicate devices") are automatically classified as Class III. These medical devices with a lower risk profile, however, may qualify for the *De Novo* process[2] instead of the PMA. Just 10% of devices regulated by the FDA are Class III devices, but they're the ones most associated with higher risk.

As the FDA says:

> "The De Novo request provides a marketing pathway to classify novel medical devices for which general controls alone, or general and special controls, provide reasonable assurance of safety and effectiveness for the intended use, but for which there is no legally marketed predicate device. De Novo classification is a risk-based classification process.
>
> "Devices that are classified into class I or class II through a De Novo classification request (De Novo request) may be marketed and used as predicates for future premarket notification [510(k)] submissions, when applicable."

[1] *https://www.fda.gov/medical-devices/overview-device-regulation/classify-your-medical-device*

[2] *https://www.fda.gov/medical-devices/premarket-submissions-selecting-and-preparing-correct-submission/de-novo-classification-request*

WHY IT'S CRUCIAL TO MITIGATE RISK IN MEDICAL DEVICES

The classification used by the FDA is largely based on risk. All electrical devices need to follow a standard known as IEC-60601 and related standards for electrical safety, but increasingly medical devices contain significant amounts of software.

The software standard most associated with the FDA process is IEC 62304. It outlines the software development life cycle process for medical devices, expressing activities and tasks necessary for the safe design and maintenance of medical device software. Software that can be an embedded or integral part of the device or the software is itself a medical device.

IEC 62304 specifies requirements for software are a subset of the requirements for a programmable electrical medical system (PEMS). This standard identifies requirements for software that are in addition to, but not incompatible with, the requirements of IEC 60601-1 [1] for PEMS.

Because a PEMS includes elements that aren't software, not all of the requirements of IEC 60601-1 for PEMS are addressed in the IEC 62304 standard. **Software is clearly a part of the verification process for the FDA, from the specification of the software requirements to the integration of the software items into a software system.** This software system is a part of a programmable electrical subsystem (PESS), which is a part of a PEMS.

No one would be very pleased if medical devices or equipment caused pain and suffering or life-threatening conditions. Sometimes they do, despite the best efforts of designers, engineers, programmers, product managers, compliance officers, and everyone else involved in complex designs.

To create the machines that help save lives and treat conditions, medical device manufacturers created medical device development life cycles, and the FDA provides guidance on how to follow its rules. Software is an increasingly important part of these newer devices.

Classifications have changed over the years for things like telehealth, which used to be Class II. In many cases, rules have been relaxed to Class I, for example, allowing for video conferencing and other telehealth related activities on common mobile phones.

On the contrary, in other cases, guidance has been strengthened. One clear example is enhancing security: FDA guidance on enhancing security combined with threats of IoT botnet takeovers of medical devices are clear about this.

There are at least two other reasons for wanting to reduce risk.

1. The financial costs of lawsuits and loss of future business from gaining a bad reputation far outweigh the cost of risk mitigation.

   The indisputable fact is, using automated software tools reduces costs, because it can be easily integrated into an Agile DevOps or waterfall build process and reduces the need for additional testing resources, which helps to shorten development and testing time.

2. On a far more personal level, software engineers, developers, product managers, release train engineers, and everyone else involved in creating medical products want their products to work and work well. No real engineer wants unfound defects, bugs, and potential disasters! **Using sophisticated software tools to perform code coverage analysis of code saves time and money and reduces risk.** Happier engineers may not be a major corporate goal, but when it happens, the company certainly benefits.

   Untested code is risky code. Analyzing code coverage to see what pieces of software are tested, and what is not tested, is crucial for achieving increasing levels of confidence that the medical device is safe and effective. Reducing risk is a primary mission of medical device/equipment engineers, and code coverage analysis does exactly that.

Anticipating and mitigating risk is required by the U.S. FDA, and therefore by the EU regulatory agencies as well.

The problem? As mentioned earlier, more and more devices contain software. As seen with other so-called mission-critical systems, they're not entirely flawless.

WHY CODE COVERAGE IS IMPORTANT

Increasingly, more function points are being delivered by software in modern medical devices, from connection to the internet to storage of and analysis of data. Development of hardware use cases now must include software use cases, and, in many instances, the number of combinations and permutations grows quite large.

How can we be sure we have test cases for all conditions? Measuring how much of the software code is really tested is the job of code coverage tools. FMEA manual process tools only get you so far. **If a piece of software is untested, it is to be untrusted. This could potentially result in unsafe and insecure medical devices.**

IEC 62304 LIFE CYCLE PROCESSES

The International Electrotechnical Commission (IEC) is an international standards organization that prepares and publishes international standards for all electrical, electronic, and related technologies, collectively known as "electrotechnology." The FDA mandates compliance with these guidelines. Interestingly, they do not specify an exact SDLC required to implement them. This means that a medical device company can use software development methodologies like waterfall, spiral, DevOps, or others. **However you choose to develop software, the FDA 510(k) process requires that this standard be implemented.** IEC 62304 requires the following processes to be implemented:

- Software development
- Software maintenance
- Problem resolution
- Risk management
- Configuration management

IEC 62304 defines the life cycle requirements for medical device software. This standard does not prescribe the name, format, or explicit content of the documentation to be produced. The standard requires documentation of tasks, but the decision of how to package this documentation is left to the user of the standard. The key point is that the FDA strongly advises, if you want their approval or clearance for market, that medical companies follow IEC 62304.

In addition to all of the other reasons to mitigate and manage risk, the IEC 62304 **does mandate** risk management. If you don't document what you have tested, how, and how much with the FDA, you open yourself up for a discussion on risk. If you haven't used code coverage tools, they might ask you, "How did you mitigate risk?"

THE ROLE SOFTWARE CODE COVERAGE PLAYS IN THE FDA GUIDANCE

In the face of ever-increasing complexity and new computer languages and technologies, the importance of automating as much of the development and testing process as possible is key to improving quality.

One of the key factors in this is testing. The struggle in testing is to write test cases that exercise and stress not only common paths, but also edge cases. As mentioned before, the FDA considers verification to be crucial.

The rules for FDA Class II clearance for market and FDA Class III are hard to figure out because the FDA has not updated its guidance in quite a few years. They don't mandate the actual software development life cycle, but they do mandate risk management. And they point to an IEC standard to do so.

What is sometimes missing is an understanding of the role that code coverage analysis tools can play. Code coverage analysis is an aid to producing test data, automating analytics, and automating some of the reporting for the state of the medical DUT (device under test). It's also a way to assure the FDA, as well as yourselves, that all the critical paths and edge conditions have been tested

FDA REQUIREMENTS

You would think that the FDA would issue clear rules regarding what kinds of analysis and especially code coverage is required for Class II and Class III devices, but sadly they don't. They offer a fairly old "General Principles of Software Validation," describing principles for validating medical device software, as well as the software used to design, develop, or manufacture medical devices. If you build a medical device or medical equipment, you must adhere to this document's requirements and guidelines. Products that get classified as Class II or Class III also need to have rigorous design, development, testing, change management, and risk analysis design controls in place.

If your software is used (incorporated into) a product, it has to be validated. If the software is the entire product itself, of course you need to show in your 510(k) submission how it was validated. Surprisingly, software used in the creation of the software must also be validated, which is why it is important to select compiler tool chains and adjacent tools that are certified for use in the development of safety-critical systems. The Parasoft C/C++test product is TÜV SÜD certified.

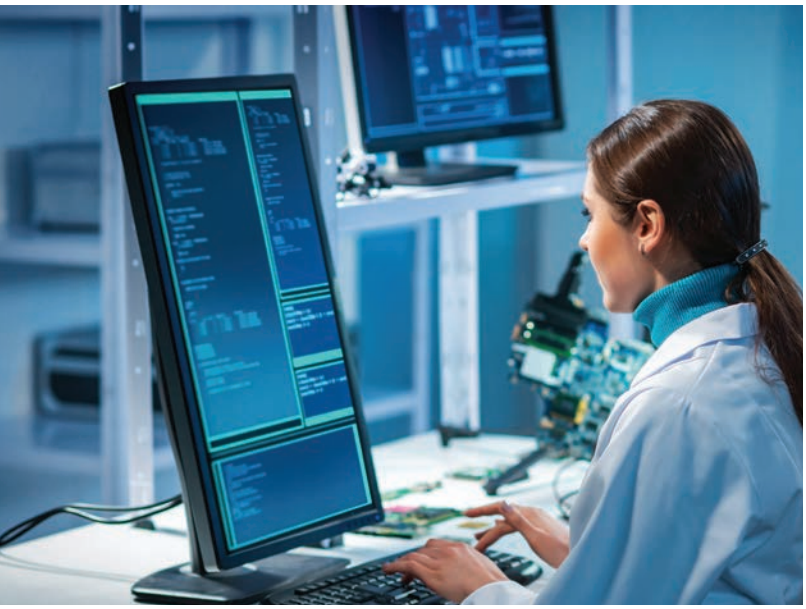
WHAT IS SOFTWARE CODE COVERAGE?

Parasoft believes that best practices as explained by the FDA include:

- **Statement coverage.** Has each statement in the program been executed? This should be a part of any FDA 510(k) Class II and III device.
- **Branch coverage.** Has each branch (also called the DD-path) of each control structure (such as in if and case statements) been executed? For example, given an if statement, have both the true and false branches been executed? (This is a subset of edge coverage). This should be a part of any FDA 510(k) Class II and III device.
- **Modified condition/decision coverage (MC/DC).** Every point of entry and exit in the program has been invoked at least once, every condition in a decision in the program has taken all possible outcomes at least once, and each condition has been shown to affect that decision outcome independently.

  A condition is shown to affect a decision's outcome independently by varying just that condition while holding fixed all other possible conditions. The condition/decision criterion does not guarantee the coverage of all conditions in the module because in many test cases, some conditions of a decision are masked by the other conditions.

  Using the modified condition/decision criterion, each condition must be shown to be able to act on the decision outcome by itself, everything else being held fixed. The MC/DC criterion is thus much stronger than the condition/decision coverage. In our experience, MC/DC code coverage metric has been widely adopted by companies doing FDA Class III products.



ADDITIONAL BENEFITS TO CODE COVERAGE

Not only are there obvious wins in increasing the confidence level of medical devices, but there are other benefits. For example, during development (software engineering) of the product, the code coverage tool can monitor which pieces of code were introduced but not tested. Monitoring changes to the code base over time increases visibility into the product.

Not only will it give the FDA confidence in your product, therefore increasing the likelihood of the granting of Clearance for Market or Approval, but it also reduces the load on development, quality, compliance, and other functions. So this is the right thing to do. It's a "yes, we should do this to improve efficiency and reduce costs" move.

AUTOMATING SOFTWARE TESTING

There's no substitute for adopting automated software testing solutions that have been used in the safety-critical space before. This is where Parasoft comes in. The embedded software in a large number of medical devices, medical equipment, and other safety-required applications has been tested by Parasoft C/C++test. This fully integrated testing solution for C/C++ software development and Parasoft DTP are certified by TÜV SÜD ensuring that these products have been tested according to IEC 62304 for both host-based and embedded target applications. Therefore, complying with the requirements of national, regional, and international regulations for functional safety.

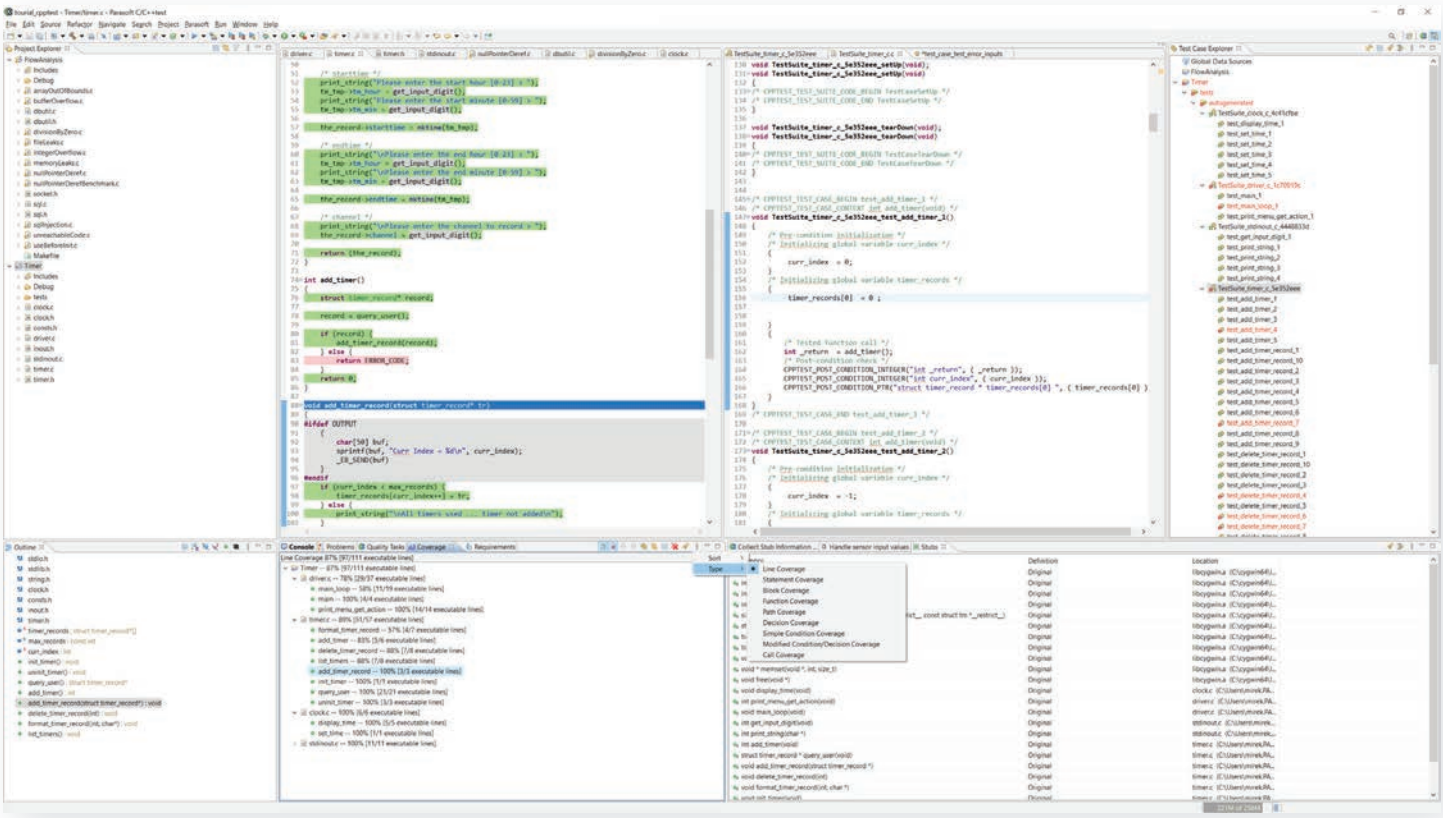
By putting in place a powerful automated testing and analytic reporting solution, software developers in test (SDETs) don't have to rely on labor consuming and error prone manual approaches. Instead, they can focus on their core development activities and increase the depth and scope of their testing to deliver high-quality software. Supporting FDA requirements and easy to use, a solution like C/C++test employs static and dynamic testing while DTP manages progress towards compliance through a central reporting and analytics dashboard.

ACTIONABLE CODE COVERAGE METRICS

Automating unit test case generation is ideal for medical device testing when using an Agile methodology like scrum and DevOps because one of the requirements is that development engineers write unit test cases. As part of the build process and before code is checked in, C/C++test automatically runs unit test cases to make sure the code is free from regressions and other bugs. In addition, with a drop-down set of configurable options available, Parasoft collects required code coverage metrics.

Teams can also run Parasoft solutions from both an integrated development environments (IDE) as well as from the command line or shell. Find the full list of supported tool chains here.

Figure 1: C/C++test green highlighted code coverage and code coverage options



Teams can view actionable analytics in any browser. DTP consolidates testing results in intelligent dashboards and detailed reports.

Figure 2:
Parasoft DTP complete code coverage dashboard

PUTTING CODE COVERAGE TOGETHER IN CI/CD FOR COMPLIANCE IN MEDICAL DEVICES

Code coverage tools like those from Parasoft can be integrated into a continuous integration/ continuous delivery pipeline so that the solution does most of the heavy lifting. Since medical devices are increasingly the target of attacks, the stresses placed on development and engineering teams trying to climb on board the DevOps train and now being told to also consider DevSecOps, are enormous.

Fortunately, Parasoft's ability to run in any number of IDEs and/or from the command line means integration is already provided and requires little to no configuration.

Parasoft's solutions can test applications running on the following:

1. Target hardware, usually an embedded board
2. Host PC
3. Host PC using a simulator

If your medical device is FDA Class III, though it requires more effort, it's recommended to test on the target hardware because the standards are so much higher for Class III. Class II can be argued as acceptable to run on the host.

MORE TECHNIQUES TO APPLY

In formal testing, unit testing might be the way to get the data easier, but it's not sufficient in and of itself. Fortunately, Parasoft allows the code merge of unit and functional and even system tests. This provides a heterogeneous and organic test scenario, which really means code coverage comes to system integration tests. Cross functional system integration tests can be tied back to your test matrix.

Here are some approaches and best practices:

- 100% code coverage can be easiest achieved via unit testing, not system testing.
- Parasoft provides the ability to merge code coverage from various test methods, manual, unit, system, and more.
- Depending on the system under test constraints, Parasoft allows you to instrument the entire code and capture coverage metrics. If no file system exists, parts of the code can be instrumented, and code coverage metrics can be merged.
- Structural coverage is defined as (statement, branch and/or MC/DC are the most common). Testing MD/DC + statement structural coverage is the most stringent approach recommended by the standard for Class C software.
- Start talking to the FDA regulator regarding code coverage earlier. It will give them a sense that you are following best practices and care about integrating all kinds of testing into your SDLC.
- Look at the predicate submissions. These are FDA 510(k) submissions made by other companies previously. If their devices or equipment does similar things as yours does, make sure your submission includes that information.

If you do an update to your software/firmware, you have to show that you have tested the update to the same level as you did for the original submission. Having a more or less automated build-unit-test to functional and system test system makes this job much easier.

Parasoft brings value to regression testing, because companies can easily reuse previously created test cases to obtain code coverage, from unit testing, through functional testing to system testing.

This alone can result in cost and time savings.

FINAL RECOMMENDATIONS

Here are five steps toward approaching implementation.

1. Define which code coverage tools to use.
2. Contact your FDA regulator, if you can, and discuss your risk mitigation strategy.
3. Integrate the tool to collect data.
4. Start defining a strategy on where you collect code coverage data, and map this to your overall testing strategy.
    - » Is it possible and feasible to collect 100% code coverage at the unit test level?
    - » Run application coverage to see what you have and see what needs to be filled in.
5. Contact Parasoft and discuss your project with one of the experts, who can help guide you further.

TAKE THE NEXT STEP

Request a demo to see how your embedded software development team can achieve code coverage requirements in Class II and Class III medical devices.

ABOUT PARASOFT

Parasoft helps organizations continuously deliver quality software with its market-proven, integrated suite of automated software testing tools. Supporting the embedded, enterprise, and IoT markets, Parasoft's technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award-winning reporting and analytics dashboard delivers a centralized view of quality enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives—security, safety-critical, Agile, DevOps, and continuous testing.