Full Spectrum Software

Time to Upgrade Your Old Codebases

Why there is no better opportunity to act than now…

If your company has very old code in one or many of your devices, there may be a perfect storm brewing. The FDA is being funded like never before. More interestingly, a large number of FDA senior staff are also White House personnel. The administration clearly intends to exert influence over the FDA, an agency which has generally operated autonomously and sometimes nearly anonymously. Major changes have begun and will continue at the FDA. It's probably a safe assumption to say there will be more, not less, regulations, rules, hurdles and scrutiny of the medical device industry than ever before.

"Comparative Effectiveness" while seemingly a reasonable concept has some scientific problems. The administration plans to conduct effectiveness studies which typically take 10 or more years to complete. Since they don't have 10 years, they have stated that they will need to use some "creative" methods to reach their empirical conclusions. As every scientist knows, the words "creative" and "empirical" do not belong in the same sentence. The question that leaps to mind is exactly how "creative" does the FDA intend to be? How creatively will they examine medical devices? As with other major issues, there have been bold statements made, but very little detail on how certain initiatives will actually be carried out.

Your ancient code bases and associated documentation may receive some highly undesirable scrutiny. That scrutiny may happen much faster than anyone expects. It would be a prudent assumption to say that time is not on your side. As a consulting firm, we know that many companies have some pretty antique code with very little documentation. Some of it is as a result of acquisitions and the documentation simply can't be found or just doesn't exist. Another reason is that the engineering focus has been on newer products. This results in documentation for older products, still working well in the field, simply not being kept up to date. There are many other reasons as well. However, if your company is in possession of an antique code base with minimal documentation, it may be prudent to think about, at the very minimum, bringing the documentation up to date. Once that process is complete, it makes sense to consider a complete re-write of the system. The reasons for a re-write include the addition of market driven features, easier maintenance and upgrades, more efficient testing, better performance and increased customer satisfaction.

The first step is to identify what you have and don't have in terms of documentation. If all you have are a Traceability Matrix and Test Protocols, then you must write the full compliment of technical specifications, including a Requirements Specification. If you have anything less than that, warning bells should be ringing loudly. You might want to consider immediately bringing in a specialized team that has strong expertise in software forensic analysis of old code and strong expertise in authoring documentation for medical devices.

If you choose to execute this process in-house, the following paragraphs should help serve as a guideline. The first step is to set up a completely “clean” machine on which you will build the system. Then identify and document all the tools required to build the system. Gather the source code and document all the steps required to execute a successful build. Catalog each tool you are using as these will have to be validated during this process.

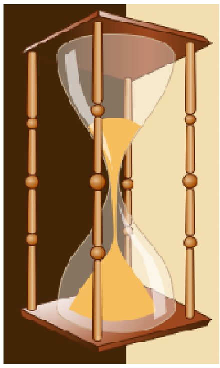
Utilize a highly senior team who are experienced in creating documentation and have them catalog all system functionality. Once they have completed the entire catalog of functionality, have them write the formal requirements. These requirements should be of sufficient quality to pass an FDA audit.

Now have the same team build Functional and Design Specifications including protocol specifications and if necessary database schema definitions. It is a best practice to tie requirements to functionality.

The next step is to perform a formal validation of all the tools used to build and test the system. Validation protocols should be well written and signed copies of the protocols should be retained. Now have the team perform a manual code review. You may choose to apply coding standards in this review as well as quality analysis. Next you should utilize static analysis and runtime analysis tools (if technically possible) to augment and verify the manual code review. While these tools can produce a huge amount of data, you should spend the necessary time and effort to tune the tools to produce high value reports.

Once these processes are complete, build traceable QA documentation for validation and verification of the system. Execute a formal V&V and ensure that all appropriate documentation is signed and dated. Create an archiving process complete with documentation on de-archiving, detailed build instructions and complete configuration instructions.

If all of this sounds time consuming and relatively unexciting from an engineering perspective, that’s because it is. It’s also an absolute requirement that each step be meticulously followed. Once you have complete documentation that could pass an FDA audit for your old system, you have reached another decision point.



Do you use the newly created documentation to develop a new system based on current software technology? The argument for doing so is extremely compelling and the benefits are undeniable.

The state of software in the medical device industry is still characterized to a larger degree than other industries by the use of very old software. There are several reasons why the medical device industry continues to maintain very old codebases.

Changes to software involve a variety of regulatory guidelines, testing and compliance issues, as we have noted above. These are very legitimate constraints and tend to hamper continuous software innovation. However, there are many more mundane and historical reasons why software innovation proceeds at a very slow pace in the medical device industry.

Hardware versus Software

Many medical device companies consider themselves to be hardware companies. In many companies, the first version of the software was written by the original team of hardware designers or research scientists who needed to get a product out the door. The code may have been written for DOS or another antiquated operating system. This old code has been updated, with great trepidation, and only when absolutely necessary. The tool sets are also old. Sometimes the databases and programming languages that the tools they were written in are literally discontinued and unsupported. The companies who produced the software tool sets have long since gone out of business or have discontinued the old technology to ensure their competitive advantages.

The notion that "If it's not broke, don't fix it" is a remarkably resilient one and still drives executive thinking. This idea combined with the time and expense of re-testing and certifying the old software for regulatory reasons provides a powerful motivation to maintain the status quo.

This state of affairs creates both business and marketing problems for the medical device industry. From a business perspective, it is difficult, time consuming and expensive to add new features and capabilities to old software.

The older software was simply never designed to handle the new features that marketing needs to maintain a competitive position in the market. This can hinder a company's ability to respond to competitive pressures in the market. Since software technology advances at a more rapid rate than hardware, the gap between hardware functionality and software capability increases quickly.

For example, it may also be difficult or impossible to translate (or localize) the software for non-English speaking markets, limiting the size of the company's total market. Old software was simply never designed to store or display non-English characters or fonts. With today's global markets, companies who cannot respond to this demand are often at a competitive disadvantage.

Can We Cost-Justify Completely Rewriting Our Software?

With the advent of new platforms and tools from Microsoft and the Linux community, the answer may be that you can't afford not to. The time to move your software to the new Microsoft or Linux platform is right now. Six months to a year from now, the FDA may have new and potentially stringent rules about very old code, there may be minimum standards that all software must meet. The simple fact is, we don't know yet.

Meanwhile your engineers will be spending months trying to patch legacy code to add a single new feature. QA will spend many more months trying to manually test whether the patches have affected other areas of the code. Their testing will have to be a manual process, since old code cannot use modern automated testing technology. This makes QA of old code exceptionally time consuming and fraught with danger. Manual testing cannot reasonably test for certain conditions. Only automated tools can perform certain tests, for example, a test that requires tens of thousands of repetitions to find a class of bug, such as an overflow.

The new Microsoft and Linux technologies provide the ability to change, modify and add functionality at a speed and with a degree of efficiency never before possible. In short, the new Microsoft and to a lesser degree Linux, platforms allow developers to create major sets of functionality simply by integrating components that exist within the platform, as opposed to writing line after line of code to create that same functionality. The difference is akin to bolting together a pre-fabricated house versus building the house in the traditional beam by beam manner. Software technology has reached a point where it is more cost effective in the long run to re-write existing software rather than applying band-aids or incrementally upgrading old software.

Highly specialized medical device software consulting firms, such as Full Spectrum Software have substantial experience working with and documenting old code. In addition, they have valuable experience in creating documentation that adheres to FDA guidelines and re-designing and re-writing older code bases using state of the art software platforms.

Conclusion

We hope you found some tips or techniques that will be beneficial to your organization. Full Spectrum Software will be hosting a series of webinars to explore each of these topics in greater detail in the near future. If you would like to be notified of upcoming webinars, please send an email to ClientServices@FullSpectrumSoftware.com.

About Full Spectrum Software

Full Spectrum Software is a 14 year old consulting firm specializing in the development of embedded and applications software for Class II and Class III medical devices. The company has delivered over 100 commercial products to market.

About the Author

Andrew Dallas, the firm's CTO, is widely considered one of the leading authorities on best practices in FDA controlled software projects Andrew serves on the Editorial Advisory Board of Medical Device and Diagnostic Industry magazine and he has published extensively in major trade and peer reviewed technical publications.

Contact Ken Carson, ClientServices@FullSpectrumSoftware.com

(508) 620-6400 • 225 Turnpike Road • Southborough, MA 01772 • www.FullSpectrumSoftware.com