

Making the Link from Requirements to Verification

Mark Pitchford

Requirements traceability is seen to yield a more predictable outcome at deployment, and responds to an increased demand for sound monitoring and management techniques during development, particularly between project phases. Appropriate automation can ensure that requirements are shown to be in compliance across the entire software development cycle, and demonstrates how a combination of tools and best practices can ease the implementation and maintenance of software development for medical devices.

Requirements Traceability

Recently, the [FDA](#) [1] took punitive action against Baxter Healthcare and their infusion pumps, eventually enforcing a recall. The [Drug and Device Accountability Act](#) [2] showed that the US government also has the issue of software quality in medical devices clearly in focus. Small wonder that medical device manufactures are looking to improve the quality of software development. Many are reviewing standards such as IEC 62304, a standard for design of medical products endorsed by the European Union and the United States, and this research has brought them face to face with a growing trend for requirements traceability.

Companies are finding that it's not enough for software to be written in a sound and robust way. It is just as important that it fulfills the requirements completely and exclusively, even if those requirements have changed during the course of the project lifecycle. The move towards requirements traceability in recent standard development acknowledges that importance.

The traditional view of software development shows each phase flowing into the next, perhaps with feedback to earlier phases, and a surrounding framework of configuration management and process (e.g., Agile, RUP). Traceability is assumed to be part of the relationships between phases. However, the reality is that while each individual phase may be conducted efficiently, the links between development tiers become increasingly poorly maintained over the duration of projects.

The answer to this conundrum lies in the Requirements Traceability Matrix (RTM), which sits at the heart of any project even if it is not identified as such (see Figure 1). Whether the links are physically recorded and managed, they still exist; for example, a developer creates a link simply by reading a design specification and using that to drive the implementation.

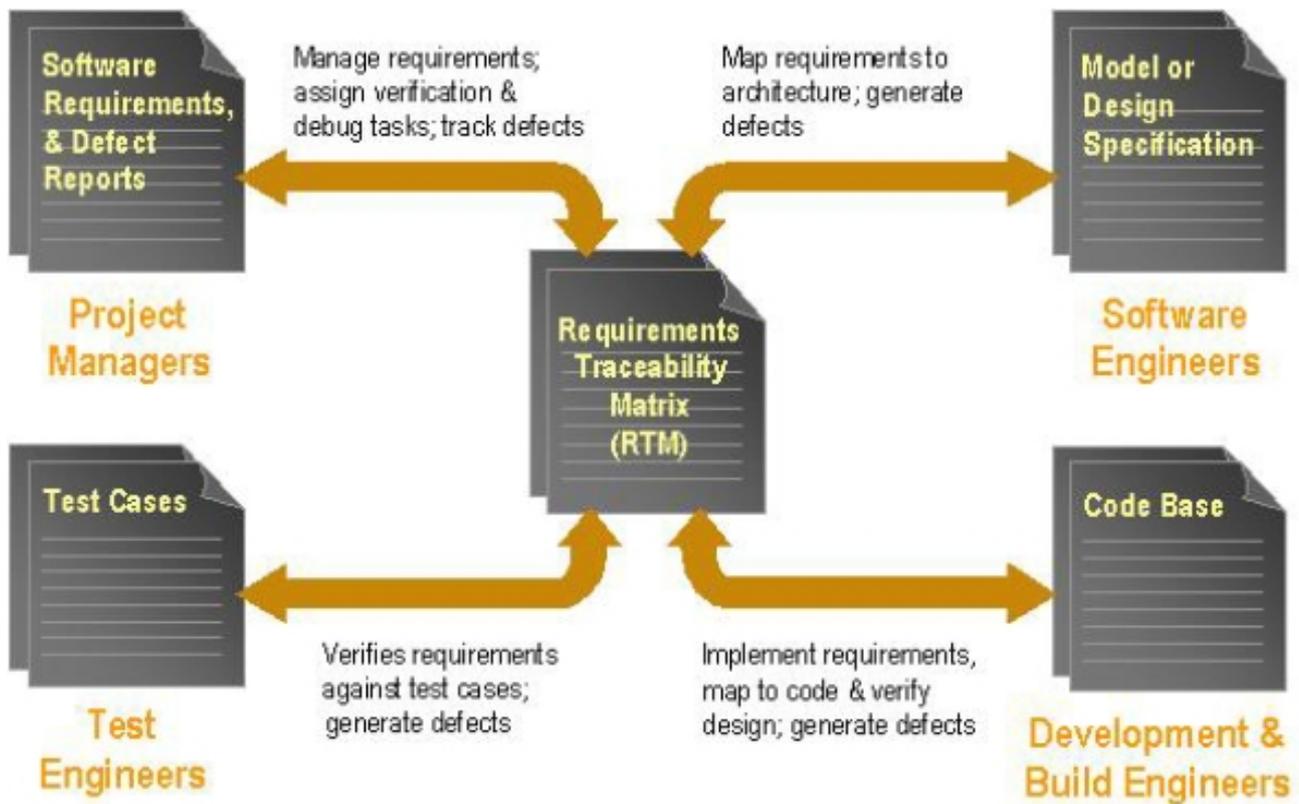


Figure 1: RTM at the heart of the project

IEC 62304 sub-clause 5.1.1 section C specifically calls for traceability to be established between system requirements, software requirements, software system test, and risk control measures implemented in software. IEC 62304 sub-clause 5.3.6 requires that steps should be taken to avoid deviation of the design from the requirement. An effectively maintained RTM can play a pivotal role in successfully meeting those requirements.

In Figure 2 the RTM is represented explicitly in the lifecycle model to emphasize its importance. With this elevated focus, the RTM can be constructed and maintained efficiently and accurately as an integral part of the development process, thereby avoiding any last-minute panic and associated costs.

- The Tier 1 high-level requirements consist of a definitive statement of the system to be developed (perhaps an automated external defibrillator (AED)) and the functional criteria it must meet (to administer an electric shock to the heart during ventricular fibrillation (VF)). This tier may be subdivided depending on the scale and complexity of the system.
- Tier 2 describes the design of the system level defined by Tier 1 and establishes links with it to begin the process of constructing the RTM. It involves the capture of low-level requirements which are specific to the design and implementation and have no impact on the functional criteria of the system. With our AED example, the low-level requirements might

Making the Link from Requirements to Verification

Published on Medical Design Technology (<http://www.mdtmag.com>)

- discuss how to monitor the patient, judge whether defibrillation is needed and then administer the shock.
- Tier 3's implementation refers to the source/assembly code developed in accordance with Tier 2. Verification activities include code rule checking and quality analysis. Maintenance of the RTM presents many challenges at this level and may involve tracing and linking requirements to individual functions. An AED involves numerous functions. Traceability of those functions back to Tier 2 requirements includes many-to-few relationships. It is very easy to overlook such relationships in a manually managed matrix.
 - In Tier 4 host-based verification, formal verification begins. Software simulation techniques help create automated test harnesses and test case generators as necessary. This tier confirms the AED functions as intended within its development environment. As reflected in IEC 62304 this is less demanding than testing on the target hardware but it allows the tests themselves to be proven quickly and efficiently.
 - Tier 5's target-based verification represents the on-target testing element of formal verification. This consists largely of confirmation that the host-based verification performed at tier 4 can be duplicated. A tier 5 RTM layer would be designed to show that the tier 4 AED tests are equally successful on the target hardware.

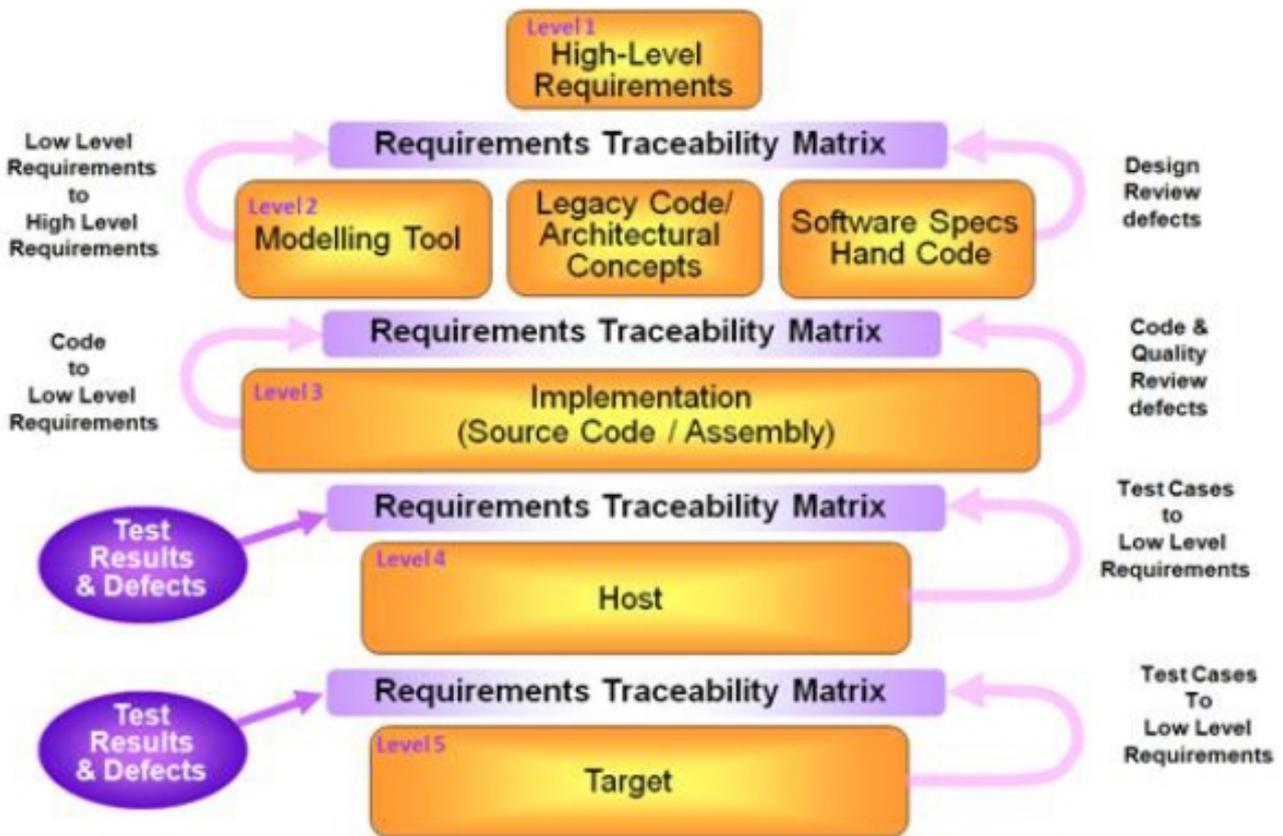


Figure 2: Development lifecycle model emphasizing the RTM

Making the Link from Requirements to Verification

Published on Medical Design Technology (<http://www.mdtmag.com>)

Gap Analysis

Many companies use gap analysis to help them evolve from current development practices to the new standard they are adopting. The wider the gap between the practices aspired to and the practices currently deployed, the greater the effort required. Happily, the scope for cost savings is similarly proportional, and gap analysis provides the necessary information for those improvements to be realized.

When current development processes are mature and tools are adequate, the greatest shortfall revealed by gap analysis typically stems from a failure to establish traceability between the software development phases so that requirements directly link to design, code, test, and verification stages.

Conclusion

Gap analysis highlights the fact that for many projects, requirements traceability remains a low-priority, manual task prone to error and omission. Consequently the construction of a requirements traceability matrix (RTM) for a medical device requiring approval can have a serious impact on costs. Tools delivering automation of the development and maintenance of an RTM can easily provide a return on investment commensurate with those automating other parts of the life cycle, be it managing requirements, supporting design, or enabling verification.

Mark Pitchford has over 25 years' experience in software development for engineering applications. He has worked on many significant industrial and commercial projects in development and management, both in the UK and internationally including extended periods in Canada and Australia. Since 2001, he has specialized in software test, and works throughout Europe and beyond as a Field Applications Engineer with LDRA Ltd.

Source URL (retrieved on 10/02/2014 - 3:25am):

http://www.mdtmag.com/articles/2012/06/making-link-requirements-verification?qt-most_popular=0

Links:

[1] <http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements>

[2] <http://www.govtrack.us/congress/bill.xpd?bill=s111-882>