Achieving Zero Defect Software Development

John Greenland, Senior Sales Manager, LDRA Technology

Zero Defect Software Development (ZDSD) is a results-oriented process that emphasizes the analysis, testing and reporting of the causality of defects. This process, which has evolved from the commitment to providing “five-nines” (99.999%) reliability for mission-critical applications, supplants the traditional approach of reacting to undesirable effects and treating symptoms while attempting to manage verification processes using “trend analyses” or simply tracking the occurrences of defects.

Under ZDSD, the first and most feasible recourse is to map system specifications and requirements to design, implementation and verification artifacts and foster requirements-based development. This process encourages the prevention and early stage detection of defects by facilitating requirements validation and verification, design verification and source code standardization.

Zero-defect software development ensures:
• higher quality and secure software products
• cost-effective software development and reduced maintenance cost

ZDSD Context Model
Depicted in Figure 1 is a context model for the verification solution proposed. This model encompasses five essential levels in embedded systems development projects, though these levels may vary significantly with respect to process and project/corporate goals, artifacts generated and overall productivity:

• Level 1: High-Level Requirements offers a definitive statement of the system to be developed and the functional criteria it must meet.
• Level 2: Low-Level Requirements involves Design and contains a representation of the design of the system characterized by Level 1. This level must establish links or traceability with Level 1 and constitutes what is called a Requirements Traceability Matrix (RTM).
• Level 3: Implementation includes the production of source code or assembly code in accordance with Level 2.
• Level 4: Host-Tier Verification is the first level dedicated to verification. At this

level, embedded software can be tested as functional and structural entities.
• Level 5: Target-Tier Verification verifies embedded software functionality that frequently crosses hardware/software boundaries making defect resolution more challenging

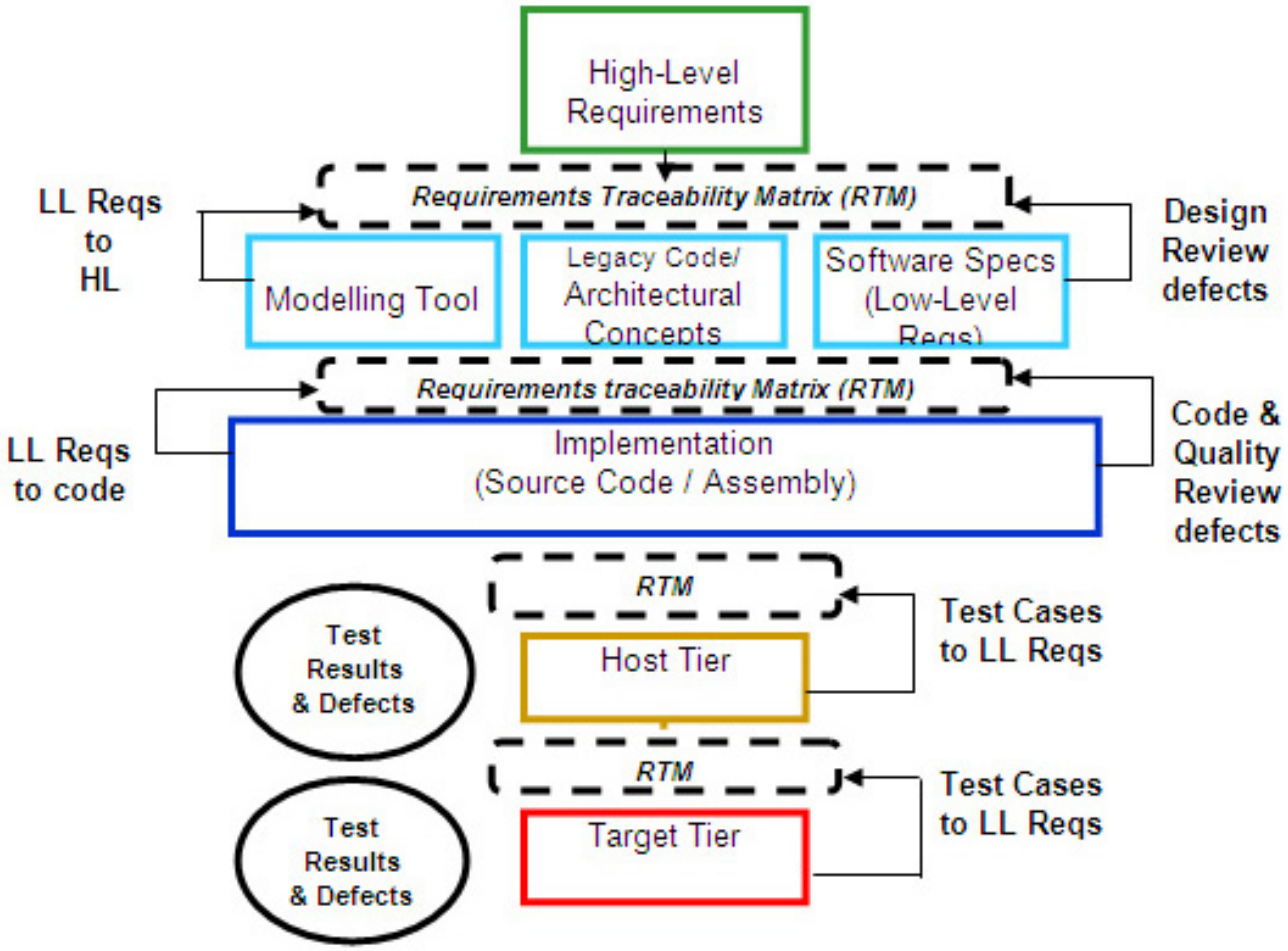

Figure 1: ZDSD context model

The ZDSD solution
The ZDSD Solution is shown in Figure 2 in the context of an iterative model for software development. A requirements traceability tool provides the Requirements Traceability Matrix (RTM) across all five levels of the ZDSD context model. This tool supports the importation of requirements from a requirements management tool or facilitates the joining of requirements into a design modeling tool.

The requirements traceability tools must include integration with a static and dynamic analysis tool that facilitates the Code Review and Quality Review in support of Level 3 (Implementation), as well as externally performed system test code coverage analysis. Additionally, the requirements traceability tool should include integration with a unit test tool that supports the software testing described at Levels 4 and 5 of the model.

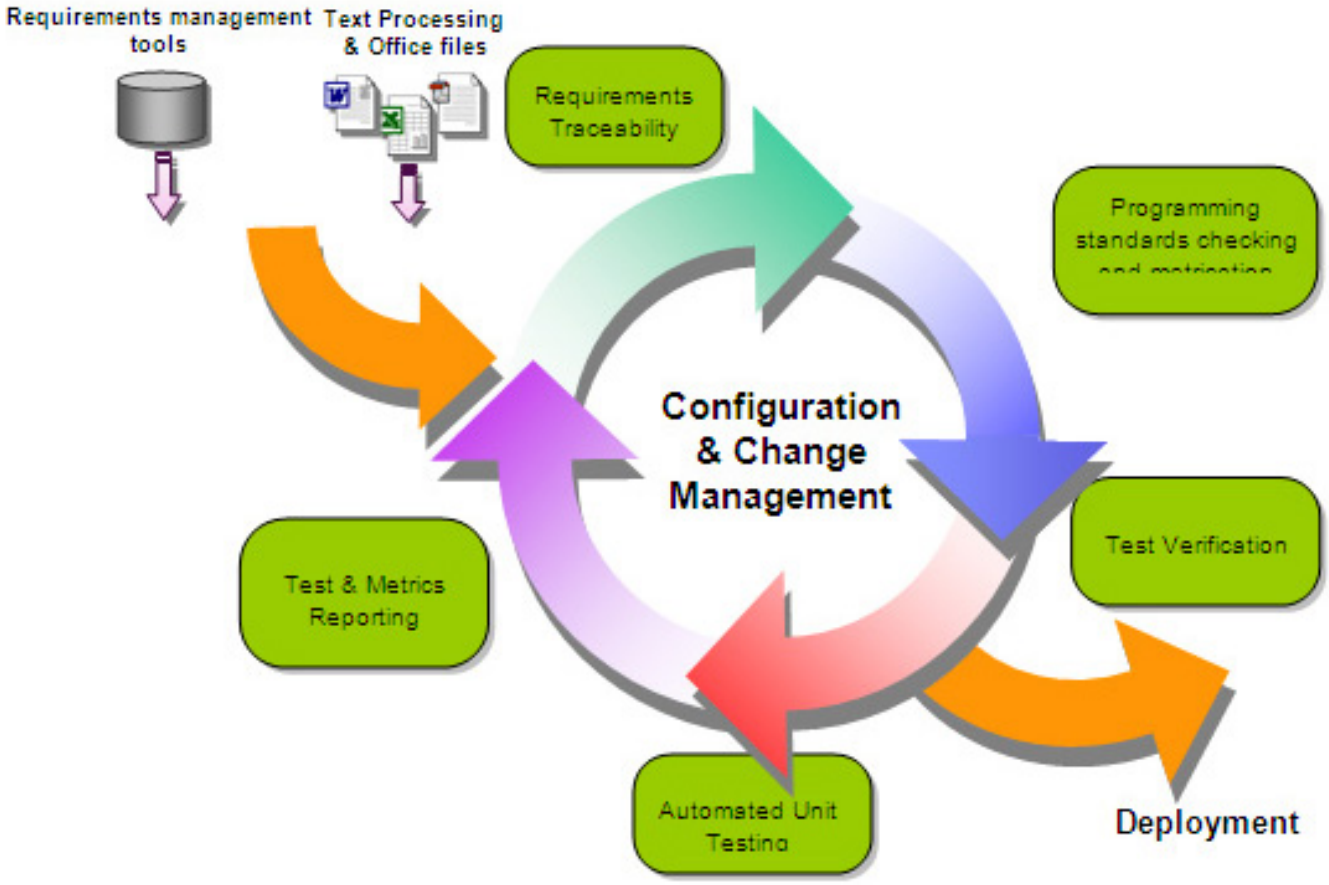

Figure 2: Automated tools applied to the development model

Conclusion
ZDSD is a largely automated yet agile process that links all the stakeholders (customers, development and verification teams and business operations) in a cohesive entity focused on measurable results. ZDSD maintains that requirements must be thoroughly and consistently verified in order to minimize, if not eradicate, defects while producing software products that add demonstrable value at the lowest possible cost over a product's lifecycle.

To refuse to implement a software quality program makes you vulnerable to software "accidents." With recent news stories about the automotive industry, all of us are freshly sensitized to the actual cost of software problems. Clearly, high-reliability software is a must because it saves lives and ensures your business is not exposed to undue financial risks.

John Greenland, Senior Sales Manager for LDRA Technology, is responsible for sales and market development in the Northeast region of the United States. John possesses over 20 years experience in the embedded software development tools industry, working in various technical, sales, and marketing roles.

Source URL (retrieved on *07/24/2014 - 1:35am*):
http://www.ecnmag.com/articles/2010/08/achieving-zero-defect-software-

development?qt-most_popular=0