Tech-Clarity

A Ten Point Guide for Streamlining Real-time Embedded Software Development

The Embedded Software Imperative

Tech-Clarity research shows that manufacturers have grown the amount of software in their products, the importance of product software, and the level of innovation driven by software over the last five years. This trend drives greater demand for embedded software development, resulting in almost three-quarters of manufacturers increasing the ratio of software engineers to other engineers. The increased prevalence, strategic importance, and resulting complexity demand that manufacturers become world-class software development and delivery companies. Tech-Clarity offers the following ten tips to help companies combat complexity and streamline embedded software development:

Integrate and Collaborate

Provide a collaborative development platform to improve software development speed, quality, and scale across large or disparate development teams. Integrate processes and systems across the entire software development lifecycle.

Keep Requirements in Focus

Track systems-level requirements as they are allocated to delivery via embedded software. Decompose the requirements to drive designs for software architecture and components, being sure not to overdesign beyond stated needs.

Define and Optimize the Architecture

Define the software architecture to support systems requirements using modular design techniques. Simulate and optimize the software architecture early in the process to improve quality and efficiency.

Align with Agile as Applicable

Adopt the appropriate level of Agile methodologies for efficiency. Balance flexibility and reduced delivery risk from Agile with business realities such as pre-defined, contractual requirements or developing software for highly regulated or safety critical applications.

Reuse Intelligently

Reuse proven software models, designs and components across products, configurations, and variants to drive higher efficiency and quality. Balance the risk of reuse with cost and an understanding of validation and verification needs for each variant.

Model to Understand and Collaborate

Develop an early understanding of software behavior by adopting model-driven development, particularly where complexity is high. Leverage a higher level of abstraction to enhance reuse and drive higher levels of productivity and

quality by eliminating early errors and taking advantage of model-based code generation.

Proactively Manage Change

Proactively understand the impact of change and ensure changes are implemented fully across the software lifecycle from requirement through testing. Integrate with systems-level change processes, coordinating change across teams and design disciplines, paying attention to the impact of changes on interfaces.

Formalize Repeatable Processes

Standardize and automate processes with best practice workflows and templates to reduce errors and improve productivity. Integrate across functions to improve coordination, avoid introducing manual errors, and ensure compliance to standards and regulations.

Enable Visibility and Traceability

Ensure that backlogs, burn down, and defects are available to the right team members in real-time, allowing collaboration across the software supply chain. Make relationships between data clear, and track everything from requirements to test results to improve quality, ensure requirements are met, enable validation and verification, more efficiently meet regulations, and increase likelihood of successfully passing audits.

Validate and Verify Early

Validate and verify early and often to maximize the value from testing and increase agility and quality. Link testing to requirements to ensure that the right functionality is tested. Use model-based testing to automate test development and execution, and integrate defect management into the change management process.

Key Takeaways

Increased importance and emphasis on embedded software is the new business reality, opening up opportunities for much more innovative, capable products but resulting in increased software development complexity. Best practices for embedded software development are emerging to streamline software development and combat complexity. A structured, integrated lifecycle approach to developing and delivering embedded software is the key to streamlining product development, driving higher levels of quality and efficiency and ensuring compliance with regulations.

About Tech-Clarity

Tech-Clarity is an independent research firm that specializes in analyzing the true business value of software technology and services.

Special Thanks to IBM

Thank you to IBM for making the distribution of this paper free.

 IBM Reference: RAL14062USEN-00