WNDRVR

THE BUSINESS CASE FOR FULL SYSTEM SIMULATION IN EMBEDDED DEVELOPMENT

ABSTRACT

When embedded system software development and testing are constrained by the availability of target hardware and systems, the entire business faces problems. Issues include slow time-to-market, high capital and operating expenses (CapEx and OpEx), suboptimal quality management, and limited security testing. The pressure to support existing embedded systems across multiple hardware platforms places further stress on development, testing, and IT operations (IT Ops) organizations. This makes DevOps, agile development, and CI/CD difficult or impossible to implement. Advances in hardware and system simulation, as exemplified by Wind River® Simics® software, have changed the entire picture. Now, with Simics, developers and testers can work in virtual labs using software-based simulations of any target system.

TABLE OF CONTENTS

INTRODUCTION

Embedded system software development and testing are often constrained by the availability of target hardware and related systemic elements like peripherals. This limitation, long viewed as an immutable rule of product development, slows down embedded systems businesses. Slow time-to-market; high capital and operating expenses (CapEx and OpEx); and suboptimal quality management leave customers unhappy. Plus, current methods only allow for a limited range of security testing.

The need to support existing embedded systems across multiple hardware platforms further stresses development, testing, and IT operations (IT Ops) organizations. And while the business may want to take advantage of new methodologies like agile, DevOps and CI/CD, the realities of developing and testing in physical labs creates a substantial impediment to making such moves.

Advanced hardware and system simulation solutions reshape this entire dynamic. Wind River Simics software enables developers and testers to work in virtual lab environments so teams can quickly build software-based simulations for any target environment they need.

This eBook explores how hardware/system simulation solves many of the business challenges arising from the reliance on physical target hardware.

Simulation enables organizations to speed up development cycles by removing the roadblocks caused by relying on physical hardware.

THE STATUS QUO

Until recently, the process of developing, testing, deploying, and supporting embedded systems relied on access to the exact hardware and extended systems required for the intended device. In this scenario, developers and product designers needed to build physical lab environments using “target hardware” to create embedded systems and write their code.

Testers needed the same setup to run tests and ensure reliability. Operations also needed access to the target hardware to put the device into the manufacturing workflow. Support engineers could only offer meaningful help to customers if they could replicate the customer environment on identical hardware, so they also needed their own lab. The cost for setting up separate labs is often prohibitive. As a result, multiple teams must coordinate access to a shared lab — but only one can use the lab at a time. This is the current situation for many embedded system makers today. The results, depicted in Figure 1, show long development timelines and associated high levels of cost and risk.

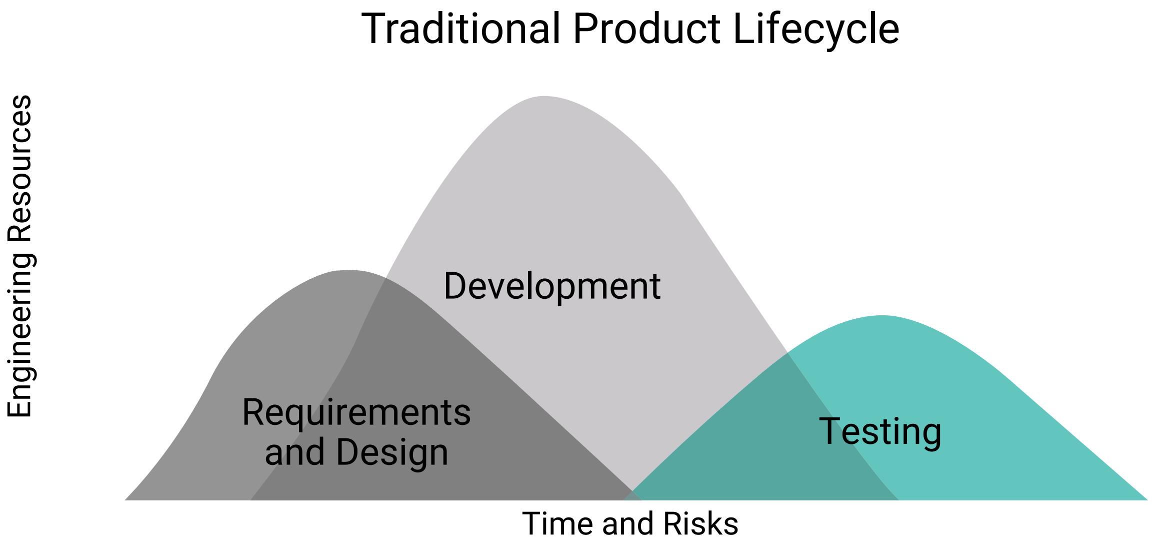


Figure 1 - With target hardware and physical labs for development and testing, the time requirements and risk inherent in creating a new embedded system are relatively high

BUSINESS PROBLEMS ARISING FROM THE STATUS QUO

The status quo is expensive and slow but it still currently works. However, it will become increasingly less viable as the industry moves to a faster, more complex product release cycle. The market expects sophisticated new embedded systems to come out on an accelerated schedule. Unfortunately, the traditional use of physical labs and target hardware/systems slows everything down.

Development Delays

Developers must wait for target hardware to emerge from prototype manufacturing, which delays development efforts and hinders the ability to automate the development process. Testers also have to wait for target hardware/systems to run their test sequences, delaying the test cycle. The inevitable rushed testing schedules limit the extent and duration of testing, resulting in hindered quality and security.

All this hardware is costly and requires a capital expense (CapEx). In most embedded systems organizations, everyone struggles with the scarcity of target systems. People wait in line for access to equipment. Even with the best of intentions, new hardware takes time to go through “tape up” and prototyping. The setup and configuration time lengthen the time-to-market cycle, slowing down revenue growth and negatively affecting competitive strategy.

Preventing New DevOps Methods

Support teams have to receive and then configure a lab featuring target hardware so they can mimic customer environments. The need to support embedded systems on multiple hardware platforms further compounds these already unscalable manual processes. For example, a device maker may want to create editions of a device that runs the Linux OS on an X86 chip, Windows on X86, and Linux on an Arm® chip. This need requires dev, test, and support teams to set up three separate sets of target system configurations. Maintaining hardware setups becomes more complex as the number of configurations grows.

Software development and the creation of new technology products are moving toward more agile, collaborative, and automated methods in the form of DevOps, agile methodologies, and continuous development/continuous integration (CI/CD). However, using these approaches to building embedded systems is effectively impossible with the current practice of using target hardware. Cross-functional teams will struggle to work together if they cannot easily access identically configured hardware/system instances.

For example, without sharing tools, data, and assets, it is quite challenging to debug a complex system. A tester may identify an issue, but it may hard to replicate. The result is a standoff. “It works on my end” is a common refrain in this scenario. It’s the customer who suffers, though, as the product goes to market with less quality assurance time than it required.

Tool Limitations

Most currently available tools were intended for evaluating hardware or simple code, not for debugging complex embedded systems that include multiple combinations of devices. They work well in their intended environment but fall short when used to test or design complex embedded systems. The result is delayed time-to-market, higher development costs, and lost revenue and market share.

Hindered Quality and Security

Often, the lack of hardware prevents teams from performing enough test cycles and varied scenarios to maintain quality and security unless the product delivery cycle stretches to accommodate the necessary time. Plus, some security tests have the potential to cause damage to the equipment, necessitating waiting for replacement hardware to continue testing. Delays in new product introduction are unacceptable because late product availability results in lost revenue. Companies are torn between the need to introduce new products as planned vs the potential for customer problems. Since the customer problems are only “potential” and can be fixed later if they do occur, speed to market usually wins out.

SOLUTION: HARDWARE AND SYSTEM SIMULATION ... AND BEYOND

One way to address the many varied challenges inherent in relying on physical labs for dev, test, and support is to use software to simulate hardware and system functionality. Using a full system simulator like Wind River Simics allows teams to take a different, more efficient approach by decoupling the software development process from hardware availability. Using advanced software, Simics can mimic a wide variety of hardware types and operating systems along with an extensive array of peripherals, boards, and networks.

Complex Simulation

The concept of using hardware simulation is not new, but Simics takes the capability to a higher level. While most simulators can validate hardware, most are too slow to run anything except simple code. In contrast, Simics can run a full software stack, including a full Linux environment. It can simulate systems of any size, from a single microcontroller to large-scale networks with hundreds or even thousands of nodes. Users can run the same binary (compiled software code) on Simics as they do on a physical hardware board.

Productive Modeling

Simics enables the user to model large interconnected systems. This might include multiple combinations of devices, architectures, and operating systems. It also provides the ability to simulate systems that don’t exist yet by mimicking the activities of boards and peripherals based on their design specifications. For example, it could simulate the functioning and speed of a new, not-yet-on-the-market Intel chip. In addition, Simics can create a realistic simulation of systems with multiple components, like satellites or IoT devices in the field.

Once the user has created a model of the system in Simics, he or she can simulate numerous operational scenarios with multiple tests running concurrently. For instance, the user can manipulate time running forward and backward, or perform deterministic bug recreations. Once a bug has been identified, the Simics user can easily pinpoint and recreate it as many times as necessary.

In short, companies can use Simics at all phases of the product lifecycle.

- In the **design phase**, they can experiment with different hardware setups to validate design assumptions before committing.
- In the **development phase**, they test and run software on virtual systems that perform exactly as they would in the physical world.
- In the **testing phase**, software debugging no longer requires expensive hardware setups and provides perfect control over the virtual target, to isolate problems efficiently.
- Throughout the **entire process**, developers work on the real target system with the same toolchain, libraries, operating system API, and operating system behavior.

Wind River uses Simics for its own product development. In our experience, Simics led to a 12,000% increase in test automation and makes bug fixes 90% faster.

BUSINESS IMPACT OF HARDWARE AND SYSTEM SIMULATION ON ACCESS PROBLEMS

Using a hardware/system simulation platform as powerful as Simics confers many business and technology benefits. The organization that employs Simics does not have to wait for target hardware to become available or struggle with the acquisition and configuration of the complicated systems needed to develop and test new embedded system software.

A Simics "virtual lab" allows for worldwide dev/test system availability and stability. Any required configuration can be set up quickly and made available globally. All users have access to the system whenever they need it, easily sharing the same artifacts. In practical terms, there's less hardware to house, maintain, and service. This lets Simics' users break the rules of embedded software development and "shift left" on the entire process. Figure 2 shows what these compressed and overlapping phases of product creation look like with the use of Simics.

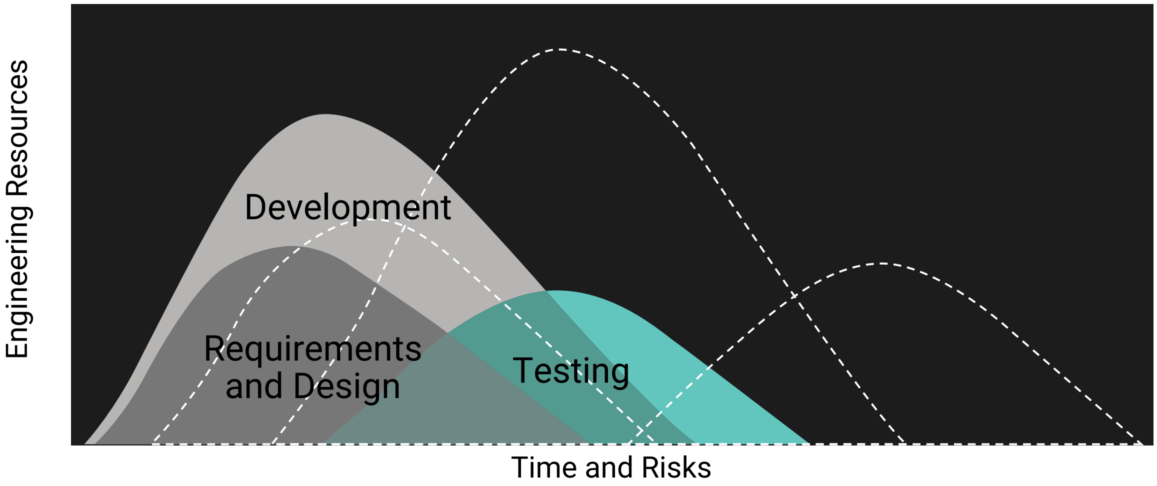


Figure 2 - Simics has proven to be able to help customers "shift left" – with requirements and design, dev, and test taking place in shorter, overlapping phases

Developer Benefits

Simics makes it easier for software development teams to adopt agile methodologies, DevOps, and CI/CD while providing better visibility of the entire development and test plan status so the teams work together toward shared goals. This potential emerges because every team member and separate team in the workflow can share identical simulated hardware in real time. Simics users have demonstrated that they can configure and reconfigure complex labs in minutes rather than weeks.

The notorious “throw it over the wall” practice that used to keep dev, test, and ops separate and uncooperative is no longer viable or necessary — because there is no more “wall.” Everyone is looking at the software running in the same environment at the same time. Teams can inject faults at any point in the process and take snapshots of process steps and results. There is no longer an excuse for not working together in an agile fashion.

INTERESTED IN LEARNING MORE ABOUT HOW WIND RIVER SOLUTIONS CAN HELP YOUR ORGANIZATION IMPLEMENT CI/CD PRACTICES?

GET THE CI/CD EBOOK

Automation Benefits

Physical hardware and system configuration are no longer bottlenecks. Development, testing, and support teams can use Simics to automate any debug, test, profile, or trace function. This enables unlimited parallelizing and scaling potential. Users can manipulate time forward and backward to isolate bugs and help find resolutions faster. They also improve time-to-market by starting or even completing test cycles even before the hardware is available.

> ***Simics users frequently reconfigure their test suites using the tool. Customers have revealed that Simics helped reduce debug time by more than 30% while simultaneously increasing quality by finding bugs sooner.***

For example, a tester can inject a fault to simulate what will happen to an embedded system when network traffic is cut off from a peripheral device. This quickly models what that looks like across different boards and configurations, ensuring more thorough testing. When a tester encounters a system failure, Simics streamlines and speeds up the process of sending the failure’s hardware and software context to developers to solve.

Quality and Security Benefits

Using some of the features already mentioned such as automation, fault injection, and time manipulation means you can test for quality and security vulnerabilities much more thoroughly than is physically possible using traditional target hardware. Using a controlled environment allows for replicable testing and delivers a consistent experience for categorizing flaws. Intentional vulnerabilities can even be injected into the system for specific fault testing. These tests can be completed before or after deployment to ensure ongoing security. Using simulation in this way ensures a positive fix and helps to avoid costly and disruptive downtime in physical environments.

> ***Plus, if something you test "destroys" the system, the fact that it's simulated means you can reset and start over instantly without expensive and risky system damage.***

Collaboration Benefits

Users find it easier to work together when they share a simulated system. Simics facilitates fluid collaboration by sharing, communicating, and exchanging executable data among developers, testers, and support people and their respective teams, including third parties.

Simics users have saved years of calendar time in large programs through the tool's sharing and collaboration functions.

The Simics toolset enables saving and sharing the circumstances of test failures. Users can show dev and test issues to each other instead of describing them in email. That alone saves time and reduces stress levels. People can simultaneously and dynamically view, manipulate, and debug an entire system in the simulation.

Financial Benefits

Improved collaboration, testing accuracy, and compressed time cycles collectively translate into financial benefits. One striking outcome is faster time-to-market for new embedded system products. One Simics customer found that Simics enabled his organization to get to market three to six months faster than had been possible with their earlier generation of physical lab technology. In particular, the development process raced forward because they could start software development activities before they had the hardware, or even silicon — i.e. “pre-silicon development.”

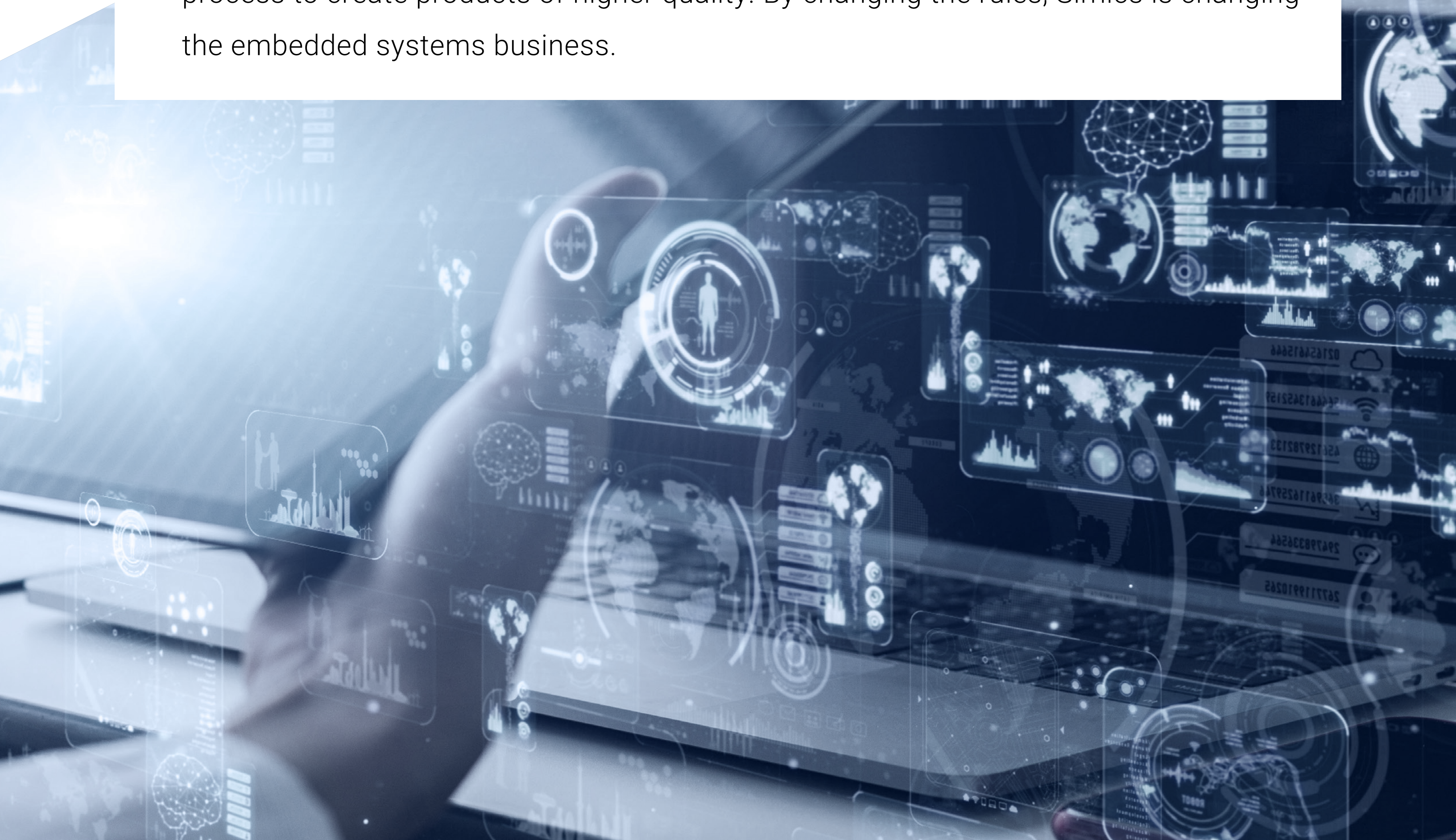
A virtual lab based on simulation is up to 93% less expensive than a physical lab and occupies less than 5% of the physical space. The latter contributes to lower facilities and utility costs for dev and test organizations. Overall, Simics enables customers to reduce CapEx and OpEx by as much as 45%.

Simics enabled users to reduce CapEx and OpEx by 45%.

CONCLUSION

The rules of embedded system product development are changing. Virtual labs using hardware/simulation solutions like Wind River Simics allow developers, product designers, and testers to work in parallel with compressed time cycles. They can take advantage of faster and more agile methodologies like DevOps. Test and support teams can dig deeper into faults and puzzling system errors while still supporting an ever-broadening portfolio of system environments. Teams collaborate using one view of the system, and they can start testing sooner by decoupling the hardware and software. As a result, they accelerate the entire development cycle.

Ultimately, system simulation makes an embedded systems business more profitable. Simulation puts products in market faster, saving on development costs and related overhead, and puts products into the revenue stage more quickly. Competitive positioning improves as companies release products more quickly than their rivals. The capital investment needed to support physical labs drops significantly. Done right, the virtual lab enabled by Simics allows all participants in the development and testing process to create products of higher quality. By changing the rules, Simics is changing the embedded systems business.



ABOUT WIND RIVER

Wind River is a global leader in delivering software for the intelligent edge. The company's technology has been powering the safest, most secure devices in the world since 1981 and is found in more than 2 billion products. Wind River offers a comprehensive portfolio supported by world-class global professional services and support and a broad partner ecosystem. Wind River software and expertise are accelerating digital transformation of critical infrastructure systems that demand the highest levels of safety, security, and reliability.

TO LEARN MORE, VISIT WWW.WINDRIVER.COM.

EMERSON NASA TOSHIBA Leading Innovation