RENESAS synergy™

Rethinking TCO In A World Driven By The Internet Of Things

Total Cost Of Ownership (TCO) of an embedded system project is a financial estimate that helps buyers and owners determine their direct and indirect costs over the lifetime of the system.

The current approach to calculating the TCO of an embedded system is usually traced to research by consulting company Gartner, but the first use of the term dates back to 1929, when the manual of the American Railway Engineering Association referenced the total cost of ownership as part of its calculations about valves.

The fast-changing IoT landscape is expected to continue increasing upfront software costs, which already consume around 61% of embedded software development resources.

October 2015

What influences the TCO of embedded systems?

When applied to embedded systems, the TCO is a measure of all the direct and indirect costs incurred over the life of the system:

- identifying, ranking and selecting vendors and products
- acquiring hardware and 3rd-party software
- in-house training
- licensing fees
- custom software development
- hardware and software integration
- installation costs
- operational costs
- maintenance costs
- in-service upgrades
- costs associated with system downtime
- associated administrative and management costs
- quality assurance costs – fault analysis & isolation, corrective actions, etc.
- exit costs associated with migrating to another system at end-of-life (EOL)

In addition, TCO reflects not just the direct qualities of an embedded development (price, functionality, reliability) but also the relationship of that system to the organization's broader set of technology platforms, installed systems, skills and strategic goals.

The Internet Of Things (IoT), sometimes called the Internet Of Everything (IoE), has been heralded as the next step in the evolution of the Internet, changing the center of mass from human-to-machine to machine-to-machine interaction.

Industry research bears this out: Navigant Research estimates that the worldwide base of smart meters will triple in size to 1.1 billion units by 2020; IHS Automotive predicts over 1523 million connected automobiles by 2020; Machina Research forecasts that consumer M2M connections will exceed 7 billion by 2023; and On World even forsees over 100 million Internet-connnected lightbulbs by 2020.

The IoT promises to change the traditional TCO model by tilting the balance decisively in favor of software versus hardware, and post-deployment costs versus initial purchase price.

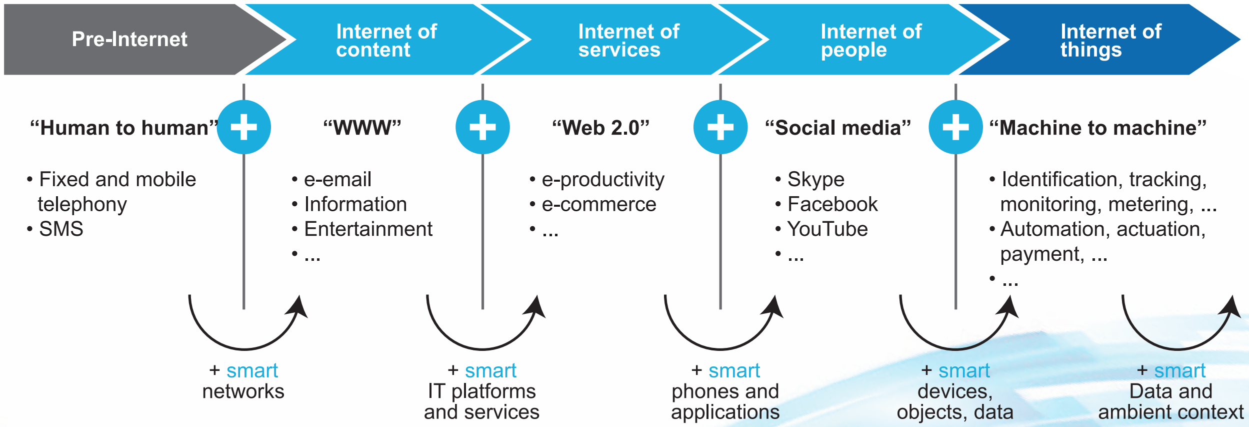


Figure 1: The road to the Internet of Things (source: Alcatel)

TCO and The Internet Of Things

Here are some key IoT trends that will change the TCO calculation over the next five years:

1. **Cheaper sensors** – High-volume applications such as wearables are driving down the ASPs of semiconductor sensors: the decline is expected to accelerate, with an annual Compound Annual Growth Rate (CAGR) of about -5% between 2015 and 2020.

2. **Cheaper bandwidth** – The cost of bandwidth has been steadily declining for a number of years: the CAGR of the median 10 Gbps price, for example, has been between -15% and -30% from 2011 - 2014.

3. **Cheaper processing** – Following Moore's Law, the number of devices on a single device has increased tremendously over the last few decades: the first microprocessor, Intel's 4004, came in at 192 transistors/mm2 compared to 8.4 million/mm2 in the current-generation Xeon Haswell-EP. At the same time, the clock speed has increased from 108kHz to as high as 4.6GHz. This has led to a corresponding decline in processing costs, which have have declined by nearly 60X in the past 10 years.

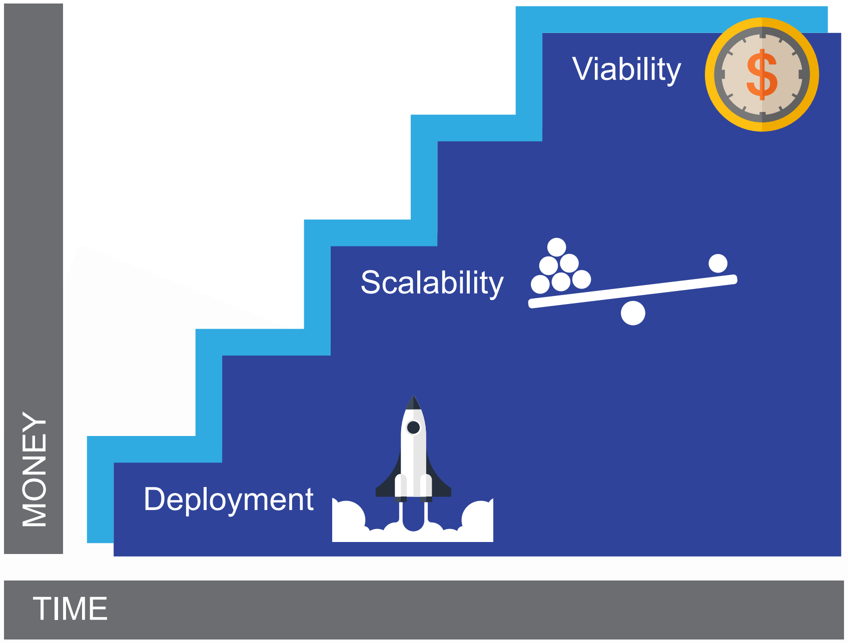
4. **A Profusion of Standards -** The IoT lacks a common set of standards and technologies that would allow for compatibility and ease-of-use. There are currently few standards (or regulations) for what is needed to run an IoT device. Industry working groups are involved in an effort to standardize the IoT, but progress has been slow.

5. **Ubiquitous wireless coverage** – Wi-Fi coverage is now ubiquitous and practically free. In addition, the adoption of the Internet Protocol version 6 (IPv6) , which supports 128-bit addresses (3.4 x 1038 devices) will allow every conceivable device to have its own unique Internet ID.

6. **Big data** – The IoT will generate enormous amounts of data, massively increasing the task of analysis; this data will migrate to the cloud for enterprise-level decision making.

The Time/Money Relationship

Will a Higher Price Pay off in Time Saved?



What are the implications for TCO calculations? The decline in purchase prices across the board – sensors, processing power, connectivity – together with the practically unlimited availability of Web addresses, will drive the adoption of intelligence to just about every node, no matter how small.

At the same time that hardware prices are declining, these trends will massively increase the complexity of the total software package in a number of areas, most notably networking, security, and maintenance. It's one thing to coordinate the operations of a single microcontroller or PLC with 100 analog inputs and a single access point; quite another to manage 100 intelligent analog access points with a variety of communication protocols that are individually Web-accessible from around the globe.

As Big Data becomes a key cloud-based management tool, it will require high-bandwidth, secure networks which can be easily updated and expanded as the volume of information grows.

How to evaluate TCO

As we've seen, the fast-changing IoT landscape will increase the upfront software costs, which already consume around 61% of embedded software development resources. The increased software complexity will also increase the proportion of TCO related to upgrades and maintenance. In the light of these trends, it's particularly important to consider a number of intangible factors.

For example, pay particular attention to likely future software technology developments. Will it be easy to add new software modules after deployment?

Scalability can be a big issue. Will you be able to upgrade to a more powerful processor without wholescale changes to the code? How about backwards compatibility with existing systems?

If your system will be in the field for many years, it's important to evaluate the long-term viability of your vendors. It may be attractive to pick a free open-source product for some complex function such as authentication, but if the vendor goes belly-up in three years, it'll be up to you to keep things together. If you're using code from a mixture of vendors (an average of 3 in the 2015 UBM Embedded Market Survey), any changes in that code may force changes in your software even if it's stable in all other respects.

In the initial phase, especially on a new project, take a close look at the software and hardware ecosystem surrounding the product.

- Does it have all the components you'll need, or will you have to develop new code for standard functions such as security or communication?
- Are all modules fully documented?
- Does the vendor provide full support for both hardware and software, or will you be farmed out to third-party vendors?
- Is the software certified if that's a requirement in your market?
- Are there consistent coding and quality standards that conform to industry best practices?
- Are there use cases and development kits that cover common IoT applications?

On average, about three-quarters of the research and development labor costs for a body of software is incurred after the software has first been shipped to paying customers as a product or part of a product. Most of the labor and cost of software is in maintenance and support, and the longer the software remains commercially viable, the higher the fraction of total effort that will go into enhancing it, fixing it and supporting it during its lifetime.

TCO and up-front costs

According to a VDC survey of developers in 2014, the size of the embedded code base is increasing at roughly three times the rate the number of embedded software developers being hired.

The result can be an uneasy blend of software from diverse sources: a company's own source and object code, externally-obtained binary executables, legacy code that may be out of date, purchased software IP, and miscellaneous other blocks of software, increasing the likelihood of low software quality, questionable reliability, and security holes that can be exploited.

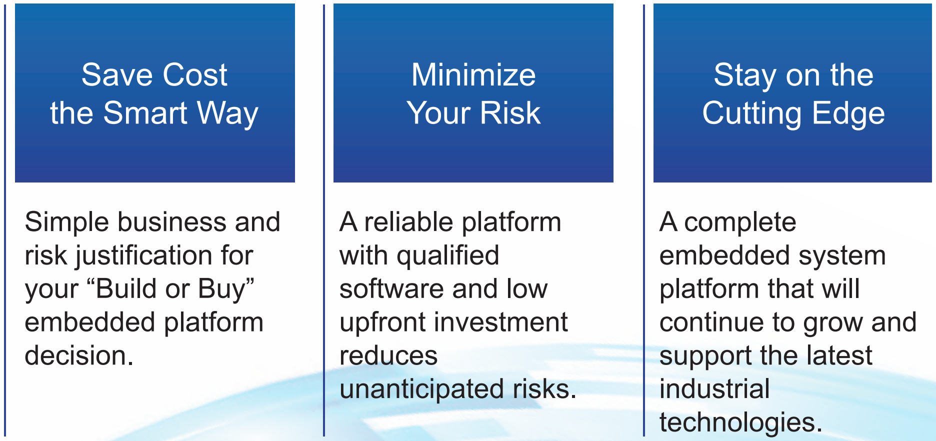
As the code size grows, so do the errors: some studies estimate that every thousand lines of code produced by commercial software developers result in 20 to 30 bugs on average.

With the fully burdened cost of a software developer being over $150,000 per year, it's a good idea to take a careful look at where spending a little more money at the start – on the software development platform, for instance – will save the use of those expensive software resources down the road.

TCO and a software development platform

An embedded software development solution provides all of the tools that developers need to do their jobs; without them, software development would screech to a halt while the team slowly assembled a collection of tools from multiple source .

A fully-integrated system that includes rigorously-tested common modules needed for IoT applications drastically reduces the development and maintenance costs over the life of the project– an important factor in the TCO. Not only that, the modules will be



Save Cost the Smart Way

Simple business and risk justification for your "Build or Buy" embedded platform decision.

Minimize Your Risk

A reliable platform with qualified software and low upfront investment reduces unanticipated risks.

Stay on the Cutting Edge

A complete embedded system platform that will continue to grow and support the latest industrial technologies.

Figure 2: TCO factors and a software development platform (Source:Renesas)

updated to incorporate improvements, bug fixes, etc., and new modules will be added as needed – critical in the IoT field, where the standards are still evolving.

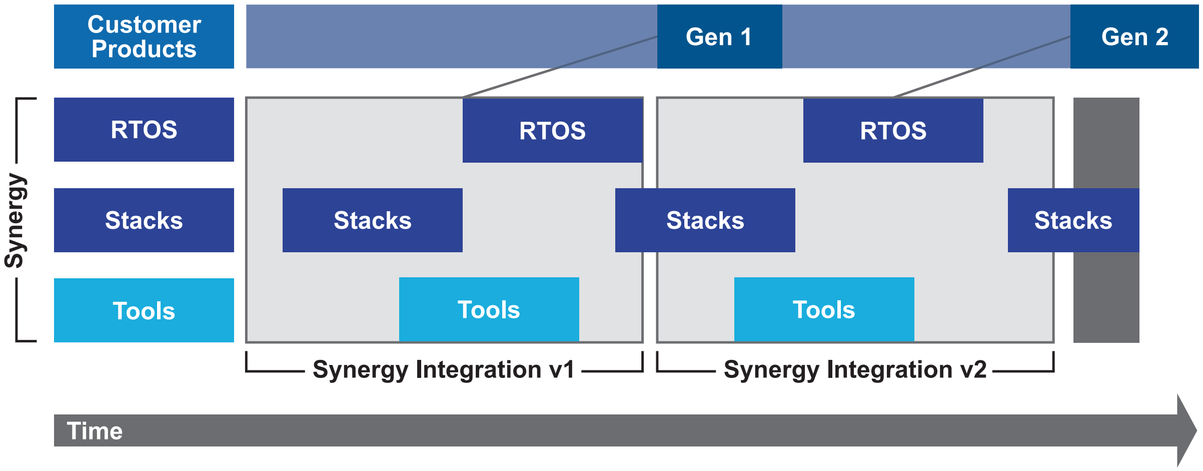
The quality and capabilities of the embedded software development solution that you select will determine your time to market and both the development cost and the manufacturing cost. It also determines the capability, performance and reliability of the end product. Given the economic realities discussed above, the cost of a high-quality software development solution is so small compared to the total development cost that it's false economy to go with a cheaper, less-capable product.

The Renesas Synergy Platform

Of course, it's important to ask the right questions when picking a software development platform. It's too easy to overlook a missing feature and have to pay a hefty price later in development, or worse, after deployment.

- Does the platform have all the components you'll need, or will you have to develop new code for standard functions such as security or communication?
- Are all modules fully documented?
- Does the vendor provide full support for both hardware and software, or will you be farmed out to third-party vendors?
- Is the software certified if that's a requirement in your market?
- Are there consistent coding and quality standards that conform to industry best practices?
- Are there use cases and development kits that cover common IoT applications?

Product development with Renesas Synergy



Like many microcontroller suppliers, we have offered software along with our hardware for many years. Our embedded system customers have told us that in the new IoT-focused environment, it's no longer enough to offer a simple suite of tools for compiling, testing and debugging code. A more comprehensive solution, with validated modules for common functions, is critical if they're to develop high-quality code and still meet the delivery date. After delivery, the development platform should include TCO-friendly tools that allow for easy maintenance and field upgrades.

The Renesas Synergy Software Platform (SSP) minimizes many of the costs inherent in TCO calculation. With Renesas Synergy, you can drastically cut your total cost of ownership with a complete platform, built upon a scalable family of microcontroller cores, that is already engineered for complex IoT technologies — optimized, integrated, tested, documented, reusable and commercially maintained over your product lifetime. With a stable, tested software development platform, your development plan is on a sound footing; your team can concentrate on developing differentiated application code, with a minimum of infrastructure changes needed between one product and the next.

Open Source versus Renesas Synergy

	Open Source	Renesas Synergy
PRICE	Open source components are usually free	Free to get started: license needed for production use (Renesas Synergy cost included in µC price)
DOCUMENTATION	Online documentation available (books may be available for some functions)	Uniform documentation standards, including context-sensitive smart manuals
EASE OF USE	Integration of modules from multiple vendors required for typical IoT application	A common API; simple integration of modules; one-stop support and consistent look & feel
SCALABILITY	Custom hardware & software development may be needed	Seamless software migration between Renesas Synergy µCs all based around the ARM Cortex-M4 family
SOFTWARE ADD-ONS	Offers a wide variety of software programs and utilities; no uniform coding standards	GUI, RTOS and numerous IoT-centric modules available (consistent coding standards)

	Open Source	Renesas Synergy
DEVELOPMENT HARDWARE	OS Software may run on standard µC development kits (level of support may be limited)	Multiple kits in development, including examples of IoT-related funcitons
QUALITY	Depends on individual vendors	Rigorous SQA program, traceability, documented coding standards, QA metrics available to customers
SECURITY	Unknown – depends on each module. Security is responsibility of the system developer	Renesas Synergy µCs include multiple security features in hardware; certified software security modules also available
SOURCE CODE	Open source modules enable users to customize or modify the code	Source code available for most modules; additional license may be required & changes may affect support
SUPPORT	Basic support is usually free via an online forum. Professional support must be purchased: quality variable	Renesas support provided for all hardware, SSP and Qualified Software modules. Support for Verified modules provided by vendors

TCO of Renesas Synergy vs. Open-Source Platforms

Of course, there are other possibilities than a ready-made commercial platform. One of the options that might be under consideration is a system built around an Open Source RTOS (free), an open source IDE such as Eclipse (also free), plus an assortment of open source software modules (also free!), then patching the whole lot together with some in-house custom software.

If you've been using such a system for several projects already and your team is familiar with its idiosyncrasies, then such an approach might seem attractive. But how does it stack up over the life of the project?

The bottom line: the open-source solution may be free, but that's just the beginning. You still have to purchase or design suitable hardware; every time you add a new module, you have to spend more time evaluating the product and the vendor, then integrate, optimize and test.

After you're in production, you'll have to keep track of new developments from multiple vendors, and repeat the integration and optimization where needed.

With Renesas Synergy, you have a range of development boards ready to go at minimal cost; the evaluation license is free; and the software platform is configured to run out of the box. Once you're in production, you'll always be able to benefit from new enhancements as they're introduced. For a new project, you maximize your code reuse, even if you need to move to another Renesas Synergy Cortex-M4 microcontroller.

Conclusion

The IoT is an exciting market, but it means that you're going to have to rethink some of the traditional TCO metrics if you want to get an accurate idea of lifetime costs.

The increased emphasis on software and post-deployment costs means that it's worth your while to pick a software development platform that will both minimize upfront development costs and allow easy upgrades to keep up with the fast-changing IoT standards.

The Renesas Synergy Platform is an ideal choice for IoT developers who want to get to market faster and minimize their TCO.

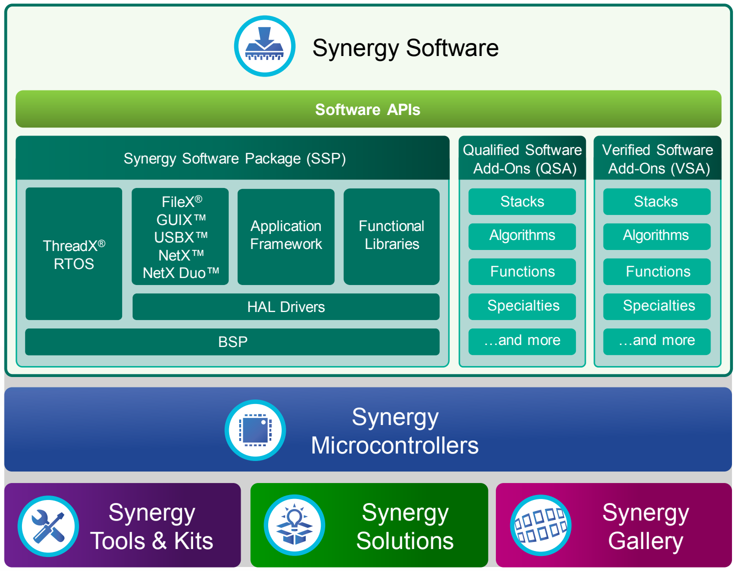
Introducing the Renesas Synergy™ Platform

Today, companies large and small race to capitalize on the rapidly growing IoT embedded system markets. Design engineers face many challenges – acquiring and mastering new technologies, developing code for low-level system infrastructure, performing integration and test, meeting aggressive schedueles – all while facing intense cost and resources pressures. A solid embedded software platform is the answer to these challenges by freeing resources to develop differentiated products instead of creating and maintaining the fundamental, yet essential system structure underneath.

Renesas introduces such a platform that is truly complete, fully tested and qualified, and systematically maintained and supported so you can start your application software development immediately at the API level without worry.

What makes the Renesas Synergy™ Platform unique?

Unlike other embedded development environments, all the Renesas Synergy™ Platform elements were designed from the ground up as a single platform. This provides unprecedented scalability and compatibility, allowing developers unparalleled code reuse. The platform will continue to grow, adding new technologies and features over time to keep your products on the cutting edge without new investments. To learn more, please visit: www.renesassynergy.com



Accelerate. Innovate. Differentiate.