Aberdeen Group
A Harte-Hanks Company

Embedded Systems Development

Three Proven Practices for Speed and Agility

March 2009

Michelle Boucher

Executive Summary

To address the growing demand for smarter products, many manufacturers are incorporating embedded systems into their products. While this approach offers exciting opportunities for innovation, it also adds a level of complexity to products that have zero tolerance for bugs and must beat the competition to market. Companies successfully addressing these challenges are reaping the benefits with higher profitability, an impressive achievement given the current economy.

Research Benchmark

Aberdeen's Research Benchmarks provide an in-depth and comprehensive look into process, procedure, methodologies, and technologies with best practice identification and actionable recommendations

Best-in-Class Performance

Aberdeen used five key performance criteria to distinguish Best-in-Class companies. Their performance on these metrics is a strong indication of product profitability and development performance. These leaders stand out from their competitors by:

- Bringing in expected product revenues 19% more often than the Industry Average and 2.9-times as often as Laggards
- Producing working software at the scheduled release date 20% more often than the Industry Average and 2.1-times as often as Laggards
- Meeting product launch targets 35% more often than the Industry Average and 3.8-times as often as Laggards

Competitive Maturity Assessment

Survey results show that the firms enjoying Best-in-Class performance shared several common characteristics. When compared to the Industry Average, they are:

- 2.3 times more likely to use virtual prototypes to validate requirements
- 92% more likely to centrally manage requirements
- 2.6 times more likely to use search and retrieval tools for reuse

"Implementing simulation tools has led to higher productivity and quality. Besides that, good tools with high user experience make developers happy, which in return increases productivity. Business-wise this should result in faster time to market and lower total-cost-of-ownership."

~ Jeroen Witteveen
Senior Systems Architect
TTOP

Required Actions

In addition to the specific recommendations in Chapter Three of this report, to achieve Best-in-Class performance, companies must:

- Develop products that meet customer needs by managing and validating design requirements throughout the development process
- Ensure software quality with a focus on testability and validation with simulation tools and source code analysis
- Accelerate the development process with integrated development tools and third party technologies

Table of Contents

Figures

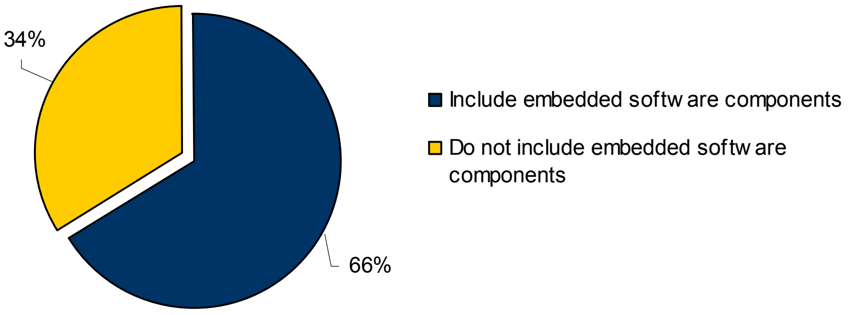
Tables

Chapter One: Benchmarking the Best-in-Class

Smart Products and Embedded Systems

Aberdeen Group's June 2008 *Engineering Executive's Strategic Agenda* identified demand for 'smarter' products as a top pressure on engineering executives today. In response, many companies are turning to embedded systems to make their products smarter. This trend is supported in research from Aberdeen Group's November 2008 *Engineering Evolved: Getting Mechatronic Performance Right the First Time* which indicates that 66% of new products developed in the last year included embedded systems (Figure 1).

Figure 1: Inclusion of Embedded System Components in Products Developed in the Past 12 Months



Source: Aberdeen Group, November 2008

This demand has made the development of embedded systems a key aspect of product development in many companies. As embedded systems become more critical, companies are naturally driven to improve how they develop them, but what specifically drives companies to seek improvement? Figure 2 displays the top five pressures participants in Aberdeen's *Embedded Systems Development* study reported as driving improvements to their process.

Top Pressures: Focus on Customers and Competition

The key drivers for improvement come from a need to grow revenues by meeting customer needs and beating the competition. Responsiveness to customer needs is represented in the top two pressures. The first - customer demand to lower the total cost of ownership - is also a driver behind the inclusion of embedded systems in the first place. Control systems that monitor power consumption of a product for example, can help keep the cost of ownership low for customers. This can be a major differentiator for manufacturers that can even justify higher price points. Improving the process for developing these systems can help yield both better performing and more profitable product.

Fast Facts

The Best-in-Class produce these results:

√ 19% more likely to meet revenue targets than the Industry Average

√ Over the last 3 years, have increased the amount of embedded software 2.6 times, 4.4 times more than their competitors

√ Release working code 22% faster than their competitors

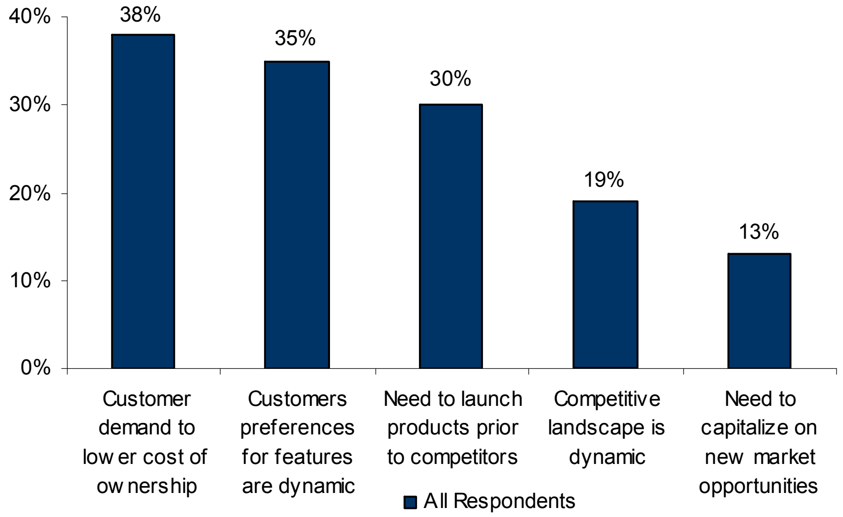
At the same time, customers are fickle. They change their minds about they want as well as which features are important to them. This means that being able to respond quickly to customer needs is critical. This drives a core question of product development: how to deliver what the customer wants, while the customer still wants it. For the professionals developing the embedded software this often means shrinking schedules, often complimented by increasing system complexity or changing scope.

Getting a product out quickly isn't just about getting it to a customer promptly; it's also about getting it out ahead of the competition. This takes shape in the third pressure reported by 30% of survey participants. Getting products out ahead of the competition enables companies to demand higher price points for new functionality before competitors can erode marketshare. However, the challenge of beating the competition to market is further complicated by the fourth pressure: a dynamic competitive landscape. This pressure is in part reflective of current economic conditions and in part reflective of the 'still emerging' status of smart products in many industries. New startups or market entrants, mergers, acquisitions, and bankruptcies make it difficult for manufacturers to identify who they are racing against.

> "For your code to work as intended you must have careful development. You need to have continuous reviews on the design and requirements. Bugs will always creep in during the design process, therefore it is important that whenever the code is developed, you need regular code inspection – at least one review meeting/inspection after a module is completed."
>
> ~Kim Fowler
> Systems Architect
> Sharfus Draid, Inc.

.

Figure 2: Top Five Pressures Driving Improvements to Embedded System Development

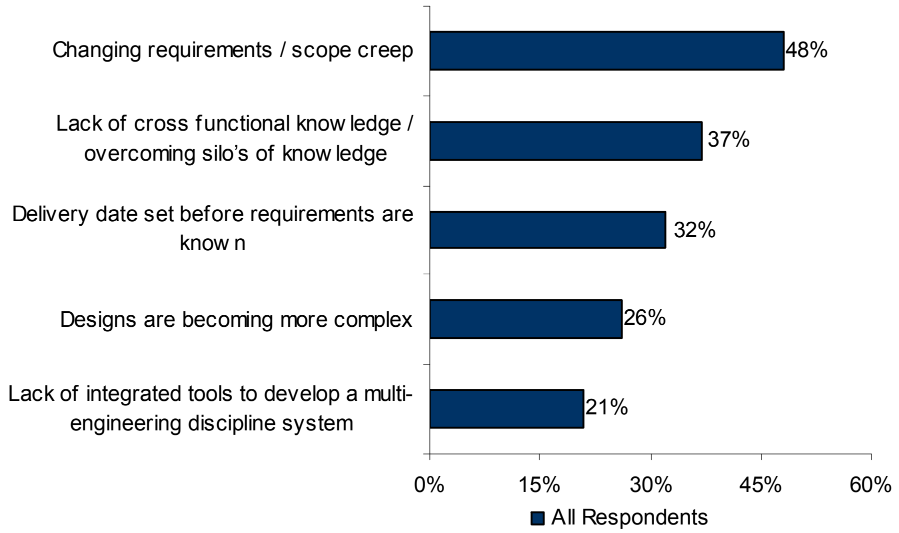


Source: Aberdeen Group, March 2009

So What Stands in the Way?

Customer and competitor-centric concerns drive improvements to embedded system development. To better understand what companies need to do to achieve profitable results, Aberdeen also asked what the most challenging aspects of developing embedded systems are. Figure 3 identifies the top five challenges.

Figure 3: Top Five Challenges Developing Embedded Systems



Source: Aberdeen Group, March 2009

Put simply: Changing requirements and the inherent complexity of products with embedded systems make the job of an embedded developer difficult. As companies are looking to add more innovation to their designs, they are looking for more out of their embedded systems. This can often happen with little regard for the status of the development project.

In fact, changing requirements was cited as the top challenge of developing embedded systems, cited by nearly half of respondents. Scope creep or new requirements added during development cycles make it difficult for developers to meet deadlines, or sufficiently validate that programs work as they should without bugs. This is related to the third challenge: project delivery dates that are set before requirements. This means that when delivery dates are set, there is little regard for the complexity of the code that will be necessary to meet the requirements.

Lack of cross functional knowledge and the challenge of overcoming siloed product development knowledge also represents a top challenge, reported by 37% of respondents. This is consistent with the top reported challenge in Aberdeen Group's January 2008 *System Design: New Product Development for Mechatronics*. While that study was more focused on system engineering aspects, the nature of the problem is largely the same. Developing integrated systems that require the collaboration of multiple engineering disciplines is inherently challenging.

The integrated nature of products with embedded systems means that firmware and software is dependant upon each other. It can be difficult to understand the impact design decisions and changes have upon other

"We're putting in more cross-functional review, so more people know what's going during development. We've always had good coupling with clinical guys and software developers. It's almost like a buddy system, but you need tools for a buddy system. We have a central repository where we will post the data that we got from the simulations– as it gets it from the virtual machine – in raw form every day. Anybody at the company can view it at any day".

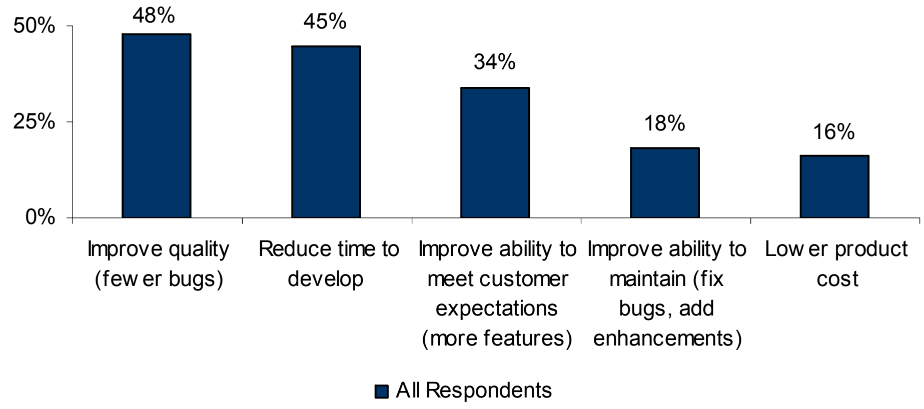
~ Chief Operating Officer
Medical Device Manufacturer

aspects of the product. Just as lack of cross-functional knowledge creates challenges for developers, a lack of tools that grant that cross-functional insight into the design creates problems as well.

…and What Needs to Be Done?

The objectives companies have in mind when making improvements to their embedded system development process, not surprisingly, align with the top pressures driving improvement: meet customer needs and beat the competition (Figure 4). In other words, the top objective (reported by 48% of respondents) is improved quality of embedded systems; the second is reduced development time, and the third is added functionality.

Figure 4: Top Five Objectives of Improvements to Embedded System Development



Source: Aberdeen Group, March 2009

Unlike other software applications, there is little tolerance for bugs in embedded software. This is especially true in safety critical applications, such as in an automobile or an airplane, where a software bug could mean loss of life. Clearly improving quality must be a top objective as customers expect quality.

While reducing bugs and ensuring the software works as intended is a top objective, getting it all done quickly is important too. Reducing the time to develop is a top objective, as well, whose importance could be deduced from top pressures to beat competitors. This objective was reported nearly as often as improving quality.

Adding more features that will appeal to customers and drive revenue is unmistakably another top objective. This objective, reported by 34% of respondents, is a response to the pressures from competitive threats and requirements to appeal to customers that are driving improvement.

Improving the ability to maintain the software and lowering the product cost are also top objectives. However, these are really secondary, falling to18% and 16% of respondents.

The Maturity Class Framework

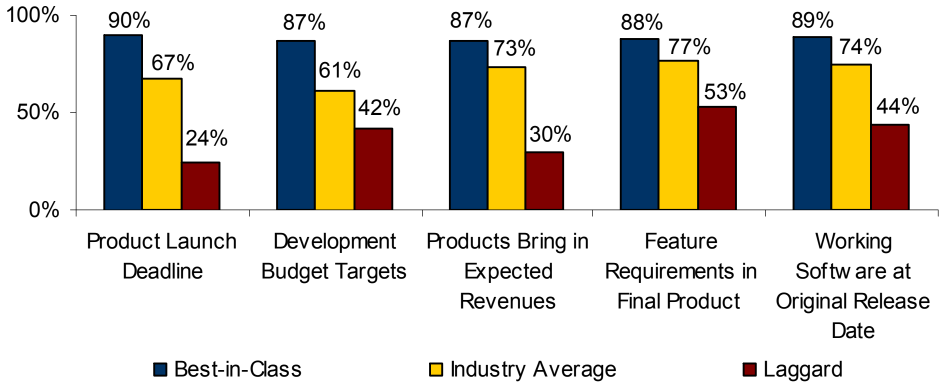
Achieving these objectives is often a matter of recognizing and successfully overcoming the challenges facing the development organization. With a clear understanding of the pressures, challenges, and objectives of embedded system developers, we can now explore the impact that best practices have on product profitability.

Aberdeen conducted survey research into embedded system development between February and March 2009. To understand what practices have the most tangible impact on organizations' ability to successfully achieve their goals, Aberdeen benchmarked respondents according to key performance criteria with a direct correlation to reported objectives. These criteria evaluated their ability to achieve the objectives they defined by assessing the percentage of products meeting the following targets:

- Product launch deadlines
- Development budgets
- Revenue Expectations
- Feature requirements included in the final product
- Working software at original release date

Using these metrics, Aberdeen classified companies into the top 20% (Best-in-Class), the middle 50% (Industry Average) and the bottom 30% (Laggard) of performers. Figure 5 displays the performance gaps that define each category.

Figure 5: The Maturity Class Framework



Source: Aberdeen Group, March 2009

With the priority given to timeliness present not just in the pressures, but the objectives and challenges as well, the performance of the Best-in-Class with regard to meet product launch deadlines is particularly striking. These performers meet these targets 3.8 times more often than Laggards. They also stay within their targeted development budgets. This is important as concessions made to achieve product launch targets often come at the expense of budgets. However, the Best-in-Class achieve both targets at nearly the same level of success.

They also meet feature requirement and working software targets with the same rate of success. This is particularly important. Not just because these can be other factors sacrificed for timeliness, but because they represent the ability to successfully address key pressures and challenges as well. Specifically, the Best-in-Class report that the requirements set out at project kickoff have been successfully implemented at the end of the project in nearly 90% of their products. By contrast, the Industry Average only accomplish this with 77% of their products; Laggards with 53%. The Best-in-Class' high score is an indication of a well executed development process that prevents cutting revenue generating features because they don't work or the project is late.

Similarly, the percentage of working software at the original scheduled release date is also a major indication of a successful development process. Aspects that aren’t working at release date are at risk for getting cut. This means that development resources go wasted on something that will not see a return. Again, the Best-in-Class meet these targets on nearly 90% of their products, saving them the difficult decision of deciding to cut features that will drive revenue or delaying the release dates and losing marketshare to competitors. It's not surprising then, that the Best-in-Class meet their revenue targets in 87% of cases as well.

Further benefits enjoyed by the Best-in-Class and that directly contribute to their high standard of success are summarized in Table 1.

> "We’ve learned to solve problems not only within the technical domain. A good tool is often an important aspect of the solution, but just introducing an architecture modeling tool won't work. Introducing a new process doesn’t work either. What will work? Introducing a solution instead of a tool (Make sure the solution includes the needs and resistance of all, direct and indirect involved people)."
>
> ~ Jeroen Witteveen
> Senior Systems Architect
> TTOP

Table 1: Performance Benefits Enjoyed by the Best-in-Class

	Best-in-Class Performance Compared to Competitors
Fewer Bugs	▪ 50% fewer bugs per lines of code ▪ 37% more likely to be satisfied with the number of bugs
Faster Development Time	▪ 2.1 times more likely to be satisfied with late project reduction ▪ 68% more likely to be satisfied with ability to meet schedules ▪ 32% faster to develop code ▪ 22% faster to release code ▪ Development time decreased by 25% in the last two years compared to an increase of 5.5% ▪ Completed projects 5 days early compared to 39 days late
Meet Customer Needs	▪ Increased the amount of embedded software in products 2.6 times over the last 3 years, 4.4 times more than competitors

Source: Aberdeen Group, March 2009

The Best-in-Class PACE Model

How do the Best-in-Class achieve such a high degree of success? Aberdeen identified a combination of strategic actions, organizational capabilities, and enabling technologies that can be summarized as follows (Table 2).

Table 2: The Best-in-Class PACE Framework

Pressures	Actions	Capabilities	Enablers
▪ Customer demand to lower cost of ownership	▪ Streamline the development process ▪ Increase use of 3rd party technologies (RTOS, databases, etc) rather than developing internally	▪ System requirements are defined using a system level block diagram ▪ Requirements are centrally managed ▪ Code simulations conducted regularly as code is written ▪ Coding standards established for the entire team ▪ Integrated environment for coding and debugging ▪ Source code is centrally managed	▪ Requirements management ▪ Automated source code analysis checks for syntax errors ▪ Performance profiling (identify performance bottlenecks) ▪ Memory leak detection ▪ Integrated environment for software modeling and coding ▪ System engineering tools ▪ Software modeling tools/block diagrams ▪ Integrated Development Environment (IDE) ▪ Search and retrieve tools for finding code components

Source: Aberdeen Group, March 2009

Best-in-Class Strategies

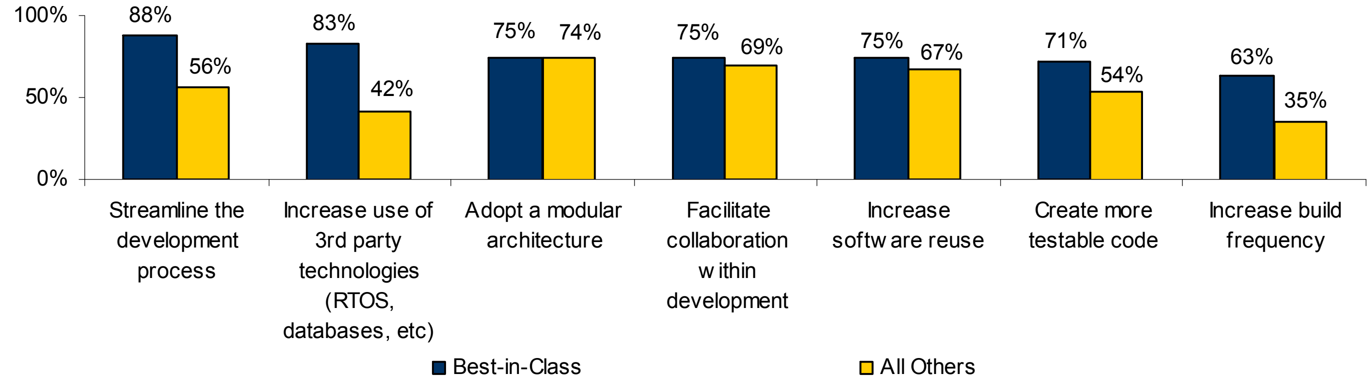
There are a number of strategies that organizations report as having a positive impact on their embedded system development process (Figure 6). Both Best-in-Class performers and their competitors are focused on adopting a modular architecture, facilitating collaboration within development, and increasing software reuse. The strategies that contribute most to Best-in-Class performance are: streamlining development processes, increased use of third party technologies, and increasing build frequency.

All of these strategies can generate improvements to embedded system development, to a greater or less degree. However, the sheer number of these strategies indicates that there is no one guarantor of Best-in-Class status. Instead, the success of these leaders has much more to do with how they execute on these strategies. In execution these strategies fall under three key themes:

- Meeting customer needs
- Ensuring program quality
- Accelerating the development process

Each of the strategies below contributes to one or more of these core principles. It is adherence to these principles that allows the Best-in-Class to meet the objectives that they set for their embedded system development initiatives and attain a higher level of performance than their peers.

Figure 6: Strategies of the Best-in-Class and their Competitors



Source: Aberdeen Group, March 2009

Aberdeen Insights — Strategy

Given the current economic situation, manufacturers must try even harder to influence buying behavior. Companies who are successful in this economic environment will be the ones whose products are not viewed as a "nice-to-have," but as a "need-to-have." Looking to embedded systems to make more innovative, smarter products is a good way to make products a "need-to-have." To be profitable means capturing what customers need to have and making sure those features are managed and enabled in the final product. It makes sense that a focus on customers is driving much of the improvements to the embedded system development process.

Aberdeen's "Tailoring Products to Customer Preferences: Configuring Profits to Order," March 2008, study examines ways to profitably configure products to customer needs. Interestingly, implementing a modular design architecture is a top Best-in-Class strategy to accomplish this. The study finds that Best-in-Class companies automate the tailoring of their products by mapping requirements to specific modules and then using that to drive the design process. They select the needed requirements and then leverage software tools to automate the configuration process.

Given the importance of focusing on customers, it makes sense that implementing a modular design architecture is a top strategy for embedded systems. By mapping specific modules to the requirement or design function they enable, it is easier to reuse existing code and offer different configurations to meet customer needs. The key is clearly defining the requirements and having tools that make it possible to take advantage of the modular architecture.

In the next chapter, we will see what these companies are doing to achieve their successful gains.

Chapter Two:
Benchmarking Requirements for Success

Competitive Assessment

Aberdeen Group analyzed the aggregated metrics of surveyed companies to determine whether their performance ranked as Best-in-Class, Industry Average, or Laggard. In addition to having common performance levels, each class also shared characteristics in five key categories: (1) **process** (the approaches they take to product development); (2) **organization** (coordination and collaboration of developers); (3) **knowledge management** (exposing product data and knowledge to key stakeholders); (4) **technology** (the tools that enhance and support the development effort); and (5) **performance management** (the ability of the organization to measure results to drive continuing improvement).

These characteristics (identified in Tables 3 - 5) serve as a guideline for best practices in embedded system development, and correlate directly with Best-in-Class performance across the key metrics. As with the strategies, these capabilities fall under three core principles:

- Meeting customer needs
- Ensuring program quality
- Accelerating the development process

Fast Facts

Top differentiators of the Best-in-Class, compared to the Industry Average:

√ 2.3 times more likely to validate requirements with virtual prototypes

√ 92% more likely to centrally manage requirements

√ 91% more likely to use automated source code analysis checks for logic errors

√ 2.6 times more likely to use search and retrieval tools for code reuse

Meeting Customer Needs: Requirements Management

Companies report meeting customer needs is a top objective for improving their embedded system development process. By aligning products with what customers want, products are more appealing, more likely to be bought, and consequently revenues will go up. It is an objective the Best-in-Class are achieving and is reflected in their ability to meet product revenue targets 19% more often than the Industry Average and 2.9 times as often as Laggards. The key to this success is capturing what customers want, defining that into product and feature requirements, and managing those requirements to ensure they make it into the final product (Table 3).

Changing requirements was reported as the top challenge of developing embedded systems, reported by 48% of respondents. This means that to be successful, companies must possess the capabilities that enable them to respond accordingly. When changes occur, those affected must be aware of the change, what the new requirement is, and be able to trace its impact on their work. The inability to successfully do this results in a number of consequences including additional bugs, extra costs, and even project delays when problems with meeting requirements are found later on during testing. Given that 'meeting customer needs' is a theme underlying three of the top pressures, the worst consequence is the inability to provide functionality customers want and as a result, expected revenues are not brought in.

"Typically there will be two different review meetings to scope out the design requirements. In the first review – this is the typically the first round of specifications, the clients will provide their specifications.

The second review will be with the key players that are defining the specifications so that it is clear. I talk to the people that qualify the equipment. I tell them what I understand, and I get an understanding of what we need from them. I always schedule the second review."

~ Cameron McDonald
Chief Executive Officer
Belfrey Automatonics

Table 3: Competitive Framework: Capabilities that Improve Ability to Meet Customer Expectations

	Best-in-Class	Industry Average	Laggard
Process	Virtual prototyping / simulation used to validate requirements have been met		
	50%	**22%**	**0%**
	Requirements traceability implemented across product development stages (design, test, etc)		
	67%	**56%**	**8%**
	Requirements traceability implemented across design functions (software, mechanical, electrical)		
	50%	**44%**	**9%**
	System requirements are defined using a system level block diagram		
	75%	**64%**	**25%**
Knowledge Management	Requirements are centrally managed		
	100%	**52%**	**42%**
Technology Enabler	Requirements management solution		
	100%	**68%**	**38%**

Source: Aberdeen Group, March 2009

To ensure products offer what customers want, companies must verify the requirements have been met. The Best-in-Class are 2.3 times more likely than the Industry Average to conduct this verification with virtual prototyping and simulations. Laggards skip this all together. By performing this validation in a virtual environment, the Best-in-Class can verify they are providing what customers need and catch gaps much earlier when it is easier to make changes, there is more time to develop code, and costs have not been invested in building target hardware.

The Best-in-Class also implement requirements traceability to track which individual requirements will meet a particular customer need. As problems are found with meeting requirements, they can trace the requirements to isolate where the problem is. In addition, as changes are made, traceability allows them to have better visibility to the impact of the change which in turn makes it easier to implement the change. This process is also supported with a system level block diagram of the requirements. Block diagrams enhance the visibility of where and how requirements are implemented, not just for software developers, but for other engineering disciplines as well. There are many interdependencies between the work of the embedded developer and that of mechanical and electrical engineers who may not follow programming. A block diagram provides a visual representation of what must be accomplished in a way that everyone can understand, regardless of engineering expertise. This provides a mechanism for addressing the number two challenge, overcoming the lack of cross functional knowledge.

Finally, the challenge of frequently changing requirements is addressed with central management of requirements. This capability is enabled with requirements management solutions. This ensures everyone is aware of the requirements and has access to them as they change.

"We put priority on finding issues as early as possible in the development cycle. This helps us reduce rounds of prototypes as well as the overall development time. Our products are fairly low volume compared to consumer type products, so covering the development cost can be a significant adder to the price of our products."

~ Mike Lease
Engineering Manager
Lancer Corporation

Ensuring Code Quality

Unlike other software applications, there is little room for error in an embedded application. Other software applications have the luxury of adding new features and providing bug fixes with software updates and patches. This is not an option for embedded software applications. Not only do customers expect their smart devices such as cell phones to work, in some applications, such as medical, automotive, and aerospace, there is a significant risk to the customer's safety if quality is not there. This message or ensuring quality is clearly reflected in the top objective to improve embedded system development processes (48% of all respondents).

Case Study: Validating Medical Devices with Simulation

Aberdeen spoke with a medical devices manufacturer of non-invasive monitoring products with a focus on measuring blood glucose levels for diabetics. As with many organizations, the company reports that their biggest source of pressure is contracting time to market cycles. However, the technology they develop is new, which means they frequently encounter performance issues that must be corrected.

To help streamline the process of finding and correcting these errors, the organization leverages virtual analysis. They create a simulated version of a device to run and validate their code. Because this organization's products are designed to measure human bio-signs, the organization tests their virtual product using human participants taking real data. This has allowed the organization to run a test in as little as four hours as opposed to the ten days they were accustomed to for clinical trial. The COO of this organization reports, 'You can get a huge amount of data in no time. We want to be able to reprocess the data in its entirety. We keep all the numbers in the screen. We can easily collect a 2GB of data a day."

This approach hasn't eliminated the need to use clinical trials. Instead, they combine simulation data and trial data to further verify the performance of their products. The ultimate impact has been decreased development time. The organization's COO reports, "Every major change can be checked out in a day instead of two weeks. We want to increase the number of times that we put the right data on the screen the first time through.

While all organizations possess little tolerance for poor quality, the Best-in-Class get it right sooner. These leaders implement their *intended* working software at original release dates 20% more often than the Industry Average and 2.1 times as often as Laggards. How do they do it? (Table 4)

Table 4: The Competitive Framework: Capabilities that Improve Quality

	Best-in-Class	Industry Average	Laggard
Process	Software written in small verifiable steps		
	75%	**68%**	**45%**
	Code is developed in modular components		
	100%	**80%**	**55%**
	Code simulations conducted regularly as code is developed		
	63%	**38%**	**17%**
	Hardware is represented during integrated simulations of embedded system		
	75%	**47%**	**25%**
Organization	Managerial visibility to bug status		
	88%	**70%**	**70%**
Knowledge Management	Coding standards established for the entire team		
	88%	**76%**	**36%**
Performance Metrics	Build over build quality analysis conducted		
	63%	**32%**	**27%**
Technology Enablers	Integrated environment for coding and debugging		
	88%	**62%**	**58%**
	Automated source code analysis checks for logic errors		
	67%	**35%**	**33%**
	Run-time analysis		
	83%	**53%**	**42%**
	Performance profiling (identify performance bottlenecks)		
	83%	**59%**	**18%**
	Memory leak detection		
	67%	**38%**	**18%**

Source: Aberdeen Group, March 2009

The Best-in-Class emphasize quality from the beginning by focusing on testability. Tight alignment with requirements allows them to focus on individual requirements, one at a time. They concentrate on small, modular sections of code that they verify works as they proceed. As a result, they are able to catch bugs on their desktop, before they make it into the build stream, 35% more often than their competitors. Ensuring the final product works, they are 3 times more likely than Laggards to conduct simulations of the integrated software and hardware. This allows them to catch problems early, when it is easiest to fix them, before building physical prototypes.

To further support quality, the Best-in-Class work in an integrated environment that enables coding and debugging at the same time. Automated checks for logic and syntax errors accelerate the testing process. The Best-in-Class also perform a variety of quality checks from performance profiling to memory leak detection, to ensure not only does the code work, but performs as required as well. Embedded software often requires an immediate response so performance tuning is critical.

By focusing on quality from the start, the Best-in-Class catch problems early in development when they can be addressed most efficiently. As a result, they avoid delays when problems are found much later when it is harder to trace the root cause and track all the areas that have been affected by the defect. This, in addition to the capabilities discussed in the next section, enable them to beat their competition to the market.

"In order to validate and make sure that our source code works as intended, we test all possible scenarios to make sure that no bugs will appear on the customer's site. In addition, we make necessary modifications to the source code and test as soon as we obtain the requirements. This way we avoid more bugs later in the development cycle."

~ Wilson Pardi Jr.
NuFlare Technology, Inc.

Accelerating the Development Process

Three of the top five pressures driving improvement to embedded system development are competitive in nature. What this means is that products must reach the market ahead of the competition, before the market develops a preference for a competitor's product.

Case Study: Belfrey Automatonics and IDE

Belfrey Automatonics is an automation and industrial controls integration house. They develop code for a wide variety of devices including: stepper and servo drives, temperature controls, remote monitoring interfaces, vision systems, pneumatics, panel wiring and other industrial devices.

They work very closely with customers to keep costs low. Cameron McDonald is the CEO and primary developer of Belfrey Automatonics. He reports, "I design for the lowest cost. I tend to work with my clients to get a clearer design for what they want – or need – or unnecessary hinges. That's how I cut down the costs for the clients. The clearer the specifications - the lower the costs. That's something that I make clear in the front end of the process - is that I need clear specifications. If it's not clear in the beginning – then you have to make changes or re-design something out."

continued

Case Study: Belfrey Automatonics and IDE

Belfrey Automatonics keeps the development process streamlined by using an integrated development environment (IDE). In the past, different methods were used programs, and code. With the IDE all this takes place in one location. McDonald states, "It's a lot easier to put all your configuration and program in one place, and keep track of all your revisions in your IDE. If I do major revisions, it's a big, big help! I am able to minimize hardware and time!"

In addition to everything else they achieve, the Best-in-Class meet product launch dates 35% more often than the Industry Average and 3.8 times as often as Laggards. The capabilities that enable this success are in Table 5.

Table 5: The Competitive Framework: Capabilities that Reduce Time to Develop

<table>
<tr><th></th><th>Best-in-Class</th><th>Industry Average</th><th>Laggard</th></tr>
<tr><td rowspan="4"><b>Knowledge Management</b></td><td colspan="3">Source code is centrally managed</td></tr>
<tr><td>100%</td><td>81%</td><td>73%</td></tr>
<tr><td colspan="3">Design data is centrally managed (including mechanical and electrical)</td></tr>
<tr><td>75%</td><td>67%</td><td>25%</td></tr>
<tr><td rowspan="4"><b>Organization</b></td><td colspan="3">Managerial visibility to project status</td></tr>
<tr><td>88%</td><td>85%</td><td>70%</td></tr>
<tr><td colspan="3">Software developers are paired up to code together (peer to peer collaboration)</td></tr>
<tr><td>50%</td><td>35%</td><td>18%</td></tr>
<tr><td rowspan="18"><b>Technology Enablers</b></td><td colspan="3">Integrated environment for software modeling and coding</td></tr>
<tr><td>50%</td><td>33%</td><td>27%</td></tr>
<tr><td colspan="3">System engineering tools</td></tr>
<tr><td>75%</td><td>63%</td><td>38%</td></tr>
<tr><td colspan="3">Software modeling tools/block diagrams</td></tr>
<tr><td>88%</td><td>45%</td><td>23%</td></tr>
<tr><td colspan="3">Integrated Development Environment (IDE)</td></tr>
<tr><td>88%</td><td>68%</td><td>46%</td></tr>
<tr><td colspan="3">Search and retrieve tools for finding code components</td></tr>
<tr><td>63%</td><td>24%</td><td>8%</td></tr>
<tr><td colspan="3">Build automation tools</td></tr>
<tr><td>88%</td><td>35%</td><td>33%</td></tr>
<tr><td colspan="3">Central repository for software assets</td></tr>
<tr><td>100%</td><td>86%</td><td>75%</td></tr>
<tr><td colspan="3">Binary Management</td></tr>
<tr><td>75%</td><td>28%</td><td>25%</td></tr>
<tr><td colspan="3">Vendor supplied RTOS</td></tr>
<tr><td>88%</td><td>70%</td><td>36%</td></tr>
</table>

Source: Aberdeen Group, March 2009

The top strategy of the Best-in-Class to improve the development of embedded systems is to streamline the development process. This starts with centralizing design information. This includes not only source code, but design data for the mechanical and electrical components that will go into the products. Centralization means everyone on the design team has access to the latest information without wasting time search for it or correcting errors due to working off of outdated information.

Managerial visibility to project status enables managers to identify potential problems with project progress that could affect completing it on time. This enables them to make adjustments to resources to prevent this from happening. In addition, Best-in-Class companies are 2.8 times more likely than Laggards to pair developers to work together as a team. This allows them to bounce ideas off each other, leading to greater innovation in less time.

The Best-in-Class leverage a variety of tools that streamline the development process. Integrated tools, whether they are software modeling tools or Integrated Development Environments (IDE), are especially powerful in enabling developers perform a variety of tasks all from within one tool, without wasting time switching between multiple applications. To support another top strategy, increase software reuse, Best-in-Class companies are 2.6 times more likely to take advantage of tools that make it easier to search and retrieve existing code. Not only does this save time because new code does not have to be written, but reusing existing code that has been previously validated saves testing time too.

Another way Best-in-Class companies are saving time is by leveraging vendor supplied Real Time Operating Systems (RTOS). They are 6.8 times more likely to use a vendor supplied RTOS than not use one. By leveraging a vendor supplied RTOS, companies can take advantage of the expertise of vendors focused specifically on an operating system designed specifically for embedded software and can instead focus on adding innovation to their own products.

"A good practice when developing embedded systems is documenting. However, when you document – you should include the rationale for your design. It is different from requirements, the "rationale of a design" explains the "why" and reasoning behind the design. Rationale is important is for future modifications and updates – people can go back and understand what you were intending to do in that design of an electrical, software, or hardware component."

~Kim Fowler
Systems Architect
Sharfus Draid, Inc.

Aberdeen Insights — Technology

Increasing the use of third party technologies such as Real Time Operating Systems (RTOS) and databases the most differentiated strategy of the Best-in-Class. Interestingly, this supports the top strategy as well, streamlining the development process. By leveraging other available third party technologies, development teams can focus their efforts on adding new innovations to their products rather than wasting efforts developing and maintaining "supporting" tools that ultimately will not play a role in the purchasing decision of their customer.

continued

Aberdeen Insights — Technology

Vendors providing RTOS, databases, interfaces, middleware and other "supporting" tools have already used their experience to perfect these tools to minimize their footprint, yet provide high-performance, real-time responsiveness. Best-in-Class companies recognize this and are more likely to take advantage of third party tools; 88% percent of them are using a vendor supplied RTOS. In contrast, 44% of their competitors report using a custom built, "home-grown" RTOS.

What is it about RTOS do Best-in-Class companies find most valuable? When asked to rank the most important characteristics of their RTOS, they report:

1. Reputation (reliability/stability) of the vendor
2. Easy to use and learn API
3. Processors supported
4. Overall performance
5. Pricing model of the vendor (licensing, upfront cost, buyout options, royalty costs)

What is interesting is that clearly it is the top strategy, streamlining the development process, that is driving what they value most in a RTOS. They want to be using tools that are easy to use and will not impact their current process or force them to change processors. What is most important to them is that once they are used to vendor's RTOS, they want to make sure that vendor will still be around so that they do not have to switch later on. These qualities are even more important than the pricing model which their competitors rate as number two, behind the impact on system resources.

Chapter Three:
Required Actions

Whether a company is trying to move its performance from Laggard to Industry Average, or Industry Average to Best-in-Class, the following actions will help spur the necessary performance improvements to develop embedded systems:

Laggard Steps to Success

- **As code is developed, validate requirements have been met with virtual prototypes.** The complexity of products with embedded systems make it difficult to predict final product behavior. In addition, target hardware is not available until late in the development cycle. Virtual prototypes provide early visibility into product behavior to ensure the product will work as defined by the requirements. None of the Laggards report doing this.
- **Conduct code simulations as the code is written.** Quality is critical with embedded systems. Validating code works as it is developed is the best time to catch bugs. Functions are fresh in the developer's mind and it is easier to trace the source of the bug. The Best-in-Class are 3.7 times more likely than Laggards to do this. Only 17% of the Laggard currently do it.
- **Leverage search and retrieval tools for finding code to reuse.** Reusing existing code speeds up the development process. Not only does it reduce time developing new code, previously validated code saves testing time too. Search and retrieval tools make it easier to find existing code. The Best-in-Class are 8.9 times more likely than Laggards to do this. Only 8% of the Laggard currently do it.

Industry Average Steps to Success

- **Centrally manage requirements.** Implementing revenue driving requirements is critical to success. Central access grants visibility of the requirements to all stakeholders. As requirements change, it becomes even harder to know what they are without central management. All of the Best-in-Class (100%) centrally manage requirements, compared to 52% of the Industry Average.
- **Leverage automated source code analysis to check for logic errors.** The Best-in-Class are 91% more likely than the Industry Average to do this. Automated source code analysis tools make it easier to identify problems and the automation saves time, providing a mechanism to implement the top object, improving quality.
- **Take advantage of search and retrieval tools to support reuse.** The Best-in-Class are 2.6 times more likely than the Industry Average to use these tools. Reducing time to develop is a top object

Fast Facts

√ Ensure products meet customer needs by managing design requirements

√ Guarantee quality by focusing on testability and validation

√ Accelerate the development process with integrated development tools and third party technologies

"The biggest impact in improving our processes for developing embedded systems were:

1) Well defined (written) coding guidelines.

2) A code review process requiring two or more different developers to provide typed/marked-up review feedback and buy-in prior to code integration into the build.

3) For extremely complex code: a sit down code review where code/designs are analyzed/changed in real-time.

4) Each developer has their own private "sand box" to develop/compile/test code prior to integration into the main code area."

~ SJ
Senior FW Engineer
Sun Microsystems

for improving embedded system development. Facilitating code reuse accomplishes this by saving development and testing time.

Best-in-Class Steps to Success

- **Increase the use of virtual prototyping to validate requirements.** Given the impact on revenue requirements have, validation is important. Predicting the behavior of complex, integrated systems is difficult. Virtual prototypes make this easier and ensure the requirements are met. Of the Best-in-Class, 50% are already taking advantage of this capability, but more could benefit.
- **Increase focus on quality metrics.** Aberdeen research has consistently shown that what is not measured can not be improved. With quality such a priority, build over build quality analyses will not only confirm quality is at the needed level, it also makes development schedules more predictable. Sixty-three percent (63%) of the Best-in-Class already do this, but increased adoption would enhance performance even further.
- **Promote teamwork to accelerate the process.** Fifty percent (50%) of the Best-in-Class pair software developers to work together. By working together, they can bounce ideas off each other and come up with innovative solutions more quickly. High performance requirements, memory constraints, and interdependent components from other engineering disciplines make embedded systems inherently complex. More Best-in-Class companies would benefit from small focused teams will solve design challenges more quickly.

Aberdeen Insights — Summary

To meet market demand for smarter products, companies are leveraging embedded systems to add innovation and "intelligence" to their products. Companies who are most successful, the Best-in-Class, are able to overcome the inherent challenges of developing products with embedded systems. They are enjoying their success with products that are 19% more likely to bring in the expected revenues when compared to the Industry Average. The practices that are key to their success are:

- Developing products that meet customer needs by managing and validating design requirements throughout the development process
- Ensuring quality by focusing on testability and validating it works with simulation tools and source code analysis
- Accelerating the development process with integrated development tools and third party technologies

With these three proven practices, the Best-in-Class are able to improve their profitability.

Appendix A: Research Methodology

Between January and March 2009, Aberdeen examined the use, the experiences, and the intentions of more than 70 enterprises that develop embedded systems. Aberdeen supplemented this online survey effort with telephone interviews with select survey respondents, gathering additional information on embedded system development strategies, experiences, and results.

Responding enterprises included the following:

- *Job title / function:* The research sample included respondents with the following job titles: engineering/development staff (40%); engineering/development manager (26%); engineering or product development director (8%); executive management (13%).
- *Industry:* The research sample included respondents primarily from both discrete and heavy manufacturing industries. The largest pools of respondents included the following sectors: industrial equipment (16%); automotive (11%); aerospace and defense (11%); high technology and electronics (23%); and medical devices (8%)
- *Geography:* The majority of respondents (59%) were from North America. Remaining respondents were from Europe (36%); the Asia-Pacific region (3%); and Middle East (2%)
- *Company size:* Twenty-nine percent (23%) of respondents were from large enterprises (annual revenues above US $1 billion); 36% were from midsize enterprises (annual revenues between $50 million and $1 billion); and 41% of respondents were from small businesses (annual revenues of $50 million or less).
- *Headcount:* Thirty-two percent (32%) of respondents were from small enterprises (headcount between 1 and 99 employees); 29% were from midsize enterprises (headcount between 100 and 999 employees); and 39% of respondents were from small businesses (headcount greater than 1,000 employees).

Solution providers recognized as sponsors were solicited after the fact and had no substantive influence on the direction of this report. Their sponsorship has made it possible for Aberdeen Group to make these findings available to readers at no charge.

Study Focus

Respondents completed an online survey that included questions designed to determine the following:

√ What is driving organizations to improve how develop embedded systems

√ The challenges they face developing embedded systems

√ The actions these companies are taking to improve how they develop embedded systems

√ The capabilities and technology enablers they have in place to support their development process

The study aimed to identify emerging best practices for the development of embedded systems and to provide a framework by which readers could assess their own capabilities.

Table 6: The PACE Framework Key

Overview

Aberdeen applies a methodology to benchmark research that evaluates the business pressures, actions, capabilities, and enablers (PACE) that indicate corporate behavior in specific business processes. These terms are defined as follows:

Pressures — external forces that impact an organization's market position, competitiveness, or business operations (e.g., economic, political and regulatory, technology, changing customer preferences, competitive)

Actions — the strategic approaches that an organization takes in response to industry pressures (e.g., align the corporate business model to leverage industry opportunities, such as product / service strategy, target markets, financial strategy, go-to-market, and sales strategy)

Capabilities — the business process competencies required to execute corporate strategy (e.g., skilled people, brand, market positioning, viable products / services, ecosystem partners, financing)

Enablers — the key functionality of technology solutions required to support the organization's enabling business practices (e.g., development platform, applications, network connectivity, user interface, training and support, partner interfaces, data cleansing, and management)

Source: Aberdeen Group, March 2009

Table 7: The Competitive Framework Key

Overview

The Aberdeen Competitive Framework defines enterprises as falling into one of the following three levels of practices and performance:

Best-in-Class (20%) — Practices that are the best currently being employed and are significantly superior to the Industry Average, and result in the top industry performance.

Industry Average (50%) — Practices that represent the average or norm, and result in average industry performance.

Laggards (30%) — Practices that are significantly behind the average of the industry, and result in below average performance.

In the following categories:

Process — What is the scope of process standardization? What is the efficiency and effectiveness of this process?

Organization — How is your company currently organized to manage and optimize this particular process?

Knowledge — What visibility do you have into key data and intelligence required to manage this process?

Technology — What level of automation have you used to support this process? How is this automation integrated and aligned?

Performance — What do you measure? How frequently? What's your actual performance?

Source: Aberdeen Group, March 2009

Table 8: The Relationship Between PACE and the Competitive Framework

PACE and the Competitive Framework – How They Interact

Aberdeen research indicates that companies that identify the most influential pressures and take the most transformational and effective actions are most likely to achieve superior performance. The level of competitive performance that a company achieves is strongly determined by the PACE choices that they make and how well they execute those decisions.

Source: Aberdeen Group, March 2009

Appendix B: Related Aberdeen Research

Related Aberdeen research that forms a companion or reference to this report include:

- *Engineering Executive's Strategic Agenda;* June 2008
- *Engineering Evolved: Getting Mechatronic Performance Right the First Time* November 2008
- *Tailoring Products to Customer Preferences: Configuring Profits to Order;* March 2008
- *System Design: New Product Development for Mechatronics;* January 2008
- *The Mechatronics Design Benchmark Report* January 2006

Information on these and any other Aberdeen publications can be found at www.Aberdeen.com.

Author: Michelle Boucher, Research Analyst, Product Innovation & Engineering Practice michelle.boucher@aberdeen.com

Since 1988, Aberdeen's research has been helping corporations worldwide become Best-in-Class. Having benchmarked the performance of more than 644,000 companies, Aberdeen is uniquely positioned to provide organizations with the facts that matter — the facts that enable companies to get ahead and drive results. That's why our research is relied on by more than 2.2 million readers in over 40 countries, 90% of the Fortune 1,000, and 93% of the Technology 500.

As a Harte-Hanks Company, Aberdeen plays a key role of putting content in context for the global direct and targeted marketing company. Aberdeen's analytical and independent view of the "customer optimization" process of Harte-Hanks (Information – Opportunity – Insight – Engagement – Interaction) extends the client value and accentuates the strategic role Harte-Hanks brings to the market. For additional information, visit Aberdeen http://www.aberdeen.com or call (617) 723-7890, or to learn more about Harte-Hanks, call (800) 456-9748 or go to http://www.harte-hanks.com