AberdeenGroup
A Harte-Hanks Company

Managing Systems Design Complexity

3 Tips to Save Time

October 2012

Michelle Boucher

Research Brief

Aberdeen Group
A Harte-Hanks Company

October 2012

Managing Systems Design Complexity: 3 Tips to Save Time

Research Brief

Aberdeen's Research Briefs provide a detailed exploration of a key finding from a primary research study, including key performance indicators, Best-in-Class insight, and vendor insight.

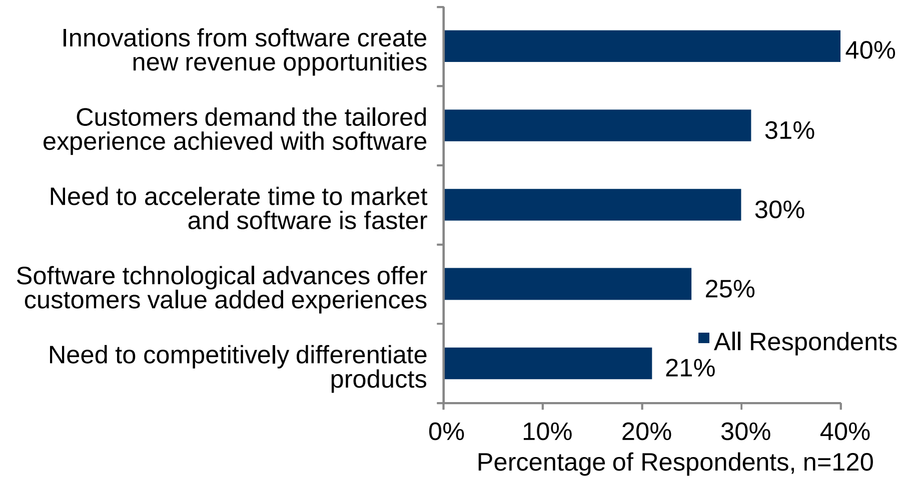
Increasingly, and in a wider array of industries, products are comprised of complex systems of interconnected software, electrical, and mechanical components. These products have grown in number, often as result of incorporating software in response to increasing market competition and customer attention to value. While the enhanced functionality offered by embedded software often results in enhanced customer value, they also cause the complexity of product systems and subsystems to grow exponentially.

Based on the experiences of organizations both incorporating embedded software and wrestling with the resulting complex systems, this report examines the strategies of managing system design complexity to identify best practices for managing requirements for complex products.

Embedded Software and Market Opportunity

Survey respondents report a 25% increase in the amount of embedded code included their products over the past two years. The reason for this expansion in embedded software lies in an uncertain economy that has seen markets grow increasingly competitive as customers become less willing to part with their money (Figure 1).

Figure 1: Top Drivers to Incorporate Embedded Software



Source: Aberdeen Group, August 2012

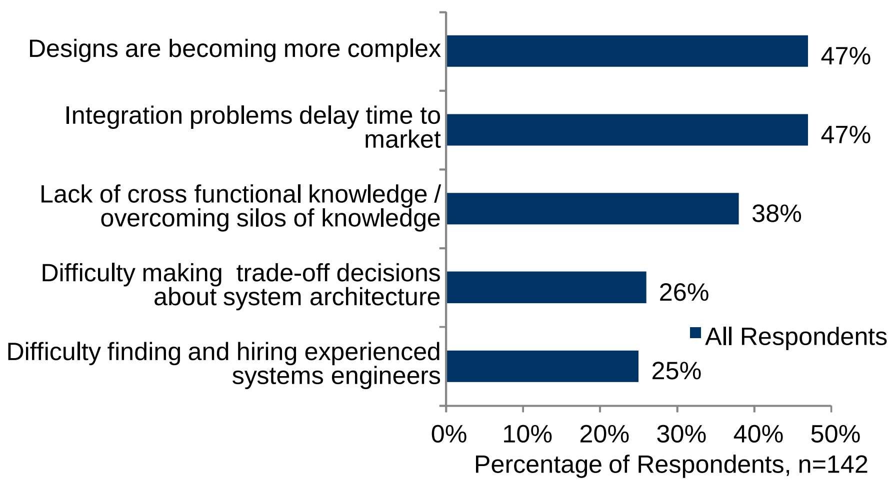
Embedded software creates new options for innovative features and functions that open doors to revenue. These features can often be developed in less time than is typically required for mechanical innovations. As such, software offers the ability to respond to customer demands for higher value in less time and at less cost. Software also offers the ability to customize a product to their preferences, creating a unique experience for each customer. The result is enhanced product differentiation for higher revenue, often at lower investments than manufacturers typically expect from product development.

The picture is not all rosy. While embedded software is becoming increasingly familiar in all sectors, it's still outside of the core competencies of many organizations. Even organizations experienced with embedded software still wrestle with incorporating these components into their products. To take advantage of the opportunities presented by embedded software, organization must improve systems engineering practices to allow for true integrations of mechanical, electrical, and software components.

The Challenge Presented by Embedded Software

In a word, the problem is: complexity (Figure 2). Software development is complex in and of itself, but incorporating it into electrical systems that must function alongside mechanical features adds a significant level of complexity.

Figure 2: Top Challenges of Designing Systems



Source: Aberdeen Group, August 2012

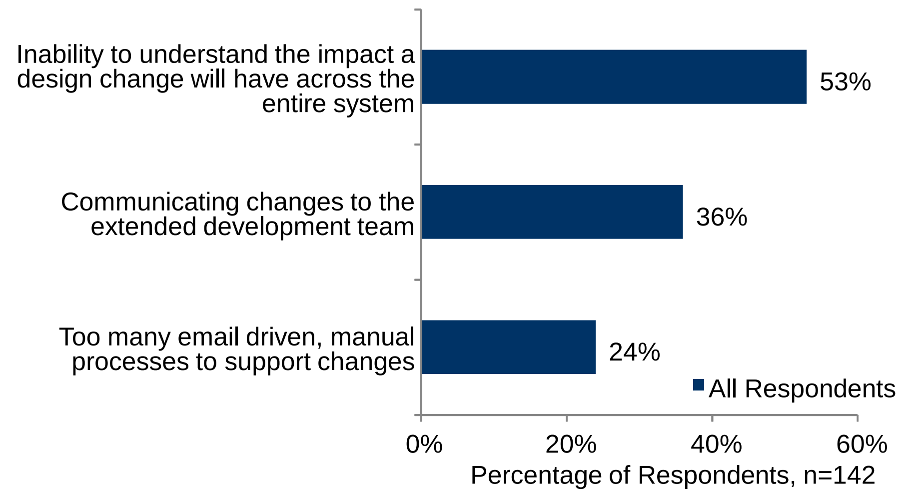
The impact is that designs become more complex. The number of engineering disciplines and the need to collaborate across them adds to this complexity. Engineers tend to be experts within their domains, but have limited understanding outside of it. The availability of experienced systems engineers continues to lag behind demand for their services. As a result, it is difficult to make system-level tradeoff decisions as well as to anticipate how the product will function once all the subsystems are brought together. Once everything comes together, systems do not always work as expected, even in the best of circumstances. This complexity and the related

Aberdeen Group
A Harte-Hanks Company

integration problems can delay product release, which impacts market competitiveness and dilutes the value that embedded software provides.

Of the many areas of product development impacted by increased complexity, few involve more painful challenges than those arise in change implementation. Expanded complexity means that it becomes more difficult to identify every aspect of the system that is affected by a change. This is the top challenge of managing system changes, reported over twice as often as manual processes (Figure 3).

Figure 3: Challenges of Managing System Changes



Source: Aberdeen Group, August 2012

Addressing this problem, as well as the myriad of other challenges raised by the complexity of system design, requires insight into interdependencies at system, subsystem, and component levels. This requires reliable processes for managing requirements as well as traceability of those requirements across the final product.

As we will see, these twin aspects of process reliability and traceability are hallmarks of the Best-in-Class. As a result, these leaders realize the revenue potential of their products.

Maturity Class Framework

To identify best practices for managing the complexity of system design, Aberdeen benchmarked the performance of study participants according to metrics indicating design efficiency and revenue generation. Aberdeen categorized participants as Best-in-Class (top 20% of performers), Industry Average (mid 50%), or Laggard (bottom 30%).

Table 1 summarizes the aggregate performance of each category.

Other Advantages of Best-in-Class Performance

In addition to the performance gains defining the Best-in-Class, Aberdeen's research revealed that these leaders have experienced the following improvements:

√ **89%** of feature requirements from project kick-off make it into the final product

√ **9% increase** in profit margins on new products

√ **13% reduction** in product cost

√ **10% reduction** in product lifecycle cost (costs of entire lifecycle from concept to "end of life," including maintenance, warranty, etc.)

√ **7% reduction** in number of change orders after release to manufacturing

√ **6% reduction** in scrap and rework costs

Telephone: 617 854 5200
Fax: 617 723 7897

Table 1: Top Performers Earn Best-in-Class Status

Definition of Maturity Class	Mean Class Performance
Best-in-Class: Top 20% of aggregate performance scorers	▪ 85% of product launch dates met ▪ 12% reduction in length of development cycle over the last 2 years ▪ 88% of product quality targets met ▪ 86% of product revenue targets met
Industry Average: Middle 50% of aggregate performance scorers	▪ 62% of product launch dates met ▪ 5% increase in length of development cycle over the last 2 years ▪ 75% of product quality targets met ▪ 68% of product revenue targets met
Laggard: Bottom 30% of aggregate performance scorers	▪ 30% of product launch dates met ▪ 13% increase in length of development cycle over the last 2 years ▪ 52% of product quality targets met ▪ 44% of product revenue targets met

Source: Aberdeen Group, August 2012

As Table 1 indicates, the Best-in-Class meet launch, quality, and revenue targets with much greater consistency than their competitors. This result directly from a superior understanding of requirements across the entire system that helps these leaders avoid integration challenges that result in time to market delays. Because the Best-in-Class get their products to market when they plan, they maximize the window of opportunity for new product revenue. The efficiency they have achieved has also permitted them to realize a 12% decrease in development time over the last two years. Both Industry Average and Laggard performers reported increases in development time as a result of their struggles with the complexity of system design.

The insight into all aspects of system and component performance helps the Best-in-Class avoid the quality challenges invited by increased product complexity. This is tied to these leaders' time-to-market advantages as well as the high quality of their released products. To this end, the Best-in-Class are 21% more likely than the Industry Average to meet quality targets at design release.

Most closely related to the market drivers of embedded software, the Best-in-Class are 20% more likely than the Industry Average to meet revenue targets. Again this results from their ability to manage the complexity of system design. The Best-in-Class possess greater control of their development processes, which helps them to ensure that more planned features are incorporated into released products. These companies do not have to cut key features to meet release deadlines or because they are not working. As a result, the products they launch incorporate the features their customers want.

Best Practices Addressing System Design Complexity

The performance of the Best-in-Class demonstrates the steps these leaders have taken to address the complexity of system design. Their approach incorporates capabilities to support:

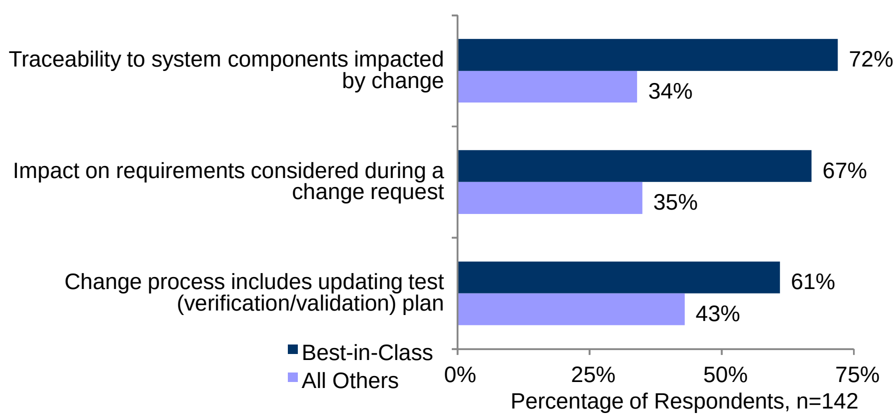
- Change Management
- Requirements Management
- Requirements Traceability

The following sections discuss each of these sets of capabilities.

Change Management

The number of components involved in designing products with embedded software makes it very difficult to identify all of the areas that are impacted by a change. This is particularly true across the multiple, often silos, engineering disciplines involved in these products. Relatedly, research from Aberdeen's *System Engineering: Top Four Design Tips to Increase Profit Margins for Mechatronics and Smart Products* (October 2009) report found that as systems become more complex, it is more difficult to ensure that design requirements are met. In response, the Best-in-Class are 2.1 times more likely than their competitors to maintain traceability across system components (Figure 4).

Figure 4: Change Management Capabilities



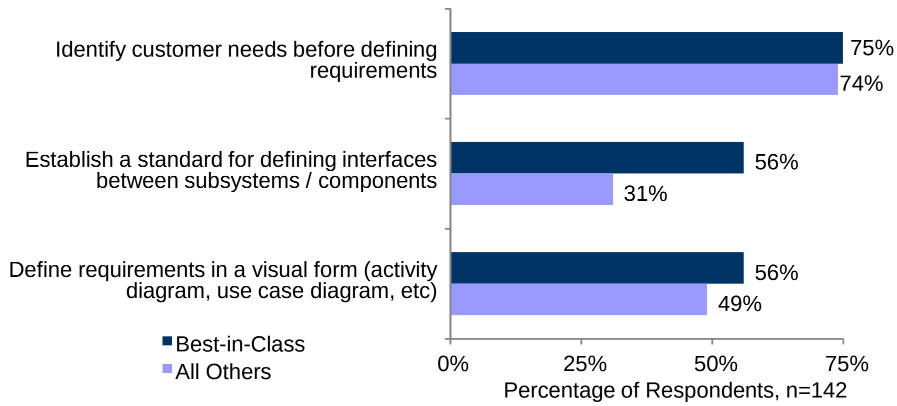
Source: Aberdeen Group, August 2012

Enhanced traceability means ongoing visibility into the relationships and interdependencies between components. This enables the Best-in-Class to track the impact of a change across the multiple components of complex systems. The Best-in-Class build on this capability by taking the step to consider the impact a requested change will have on requirements before implementing it. Without this step, engineering changes often aggravate the consequences of complexity and further muddle the design process. Instead, the Best-in-Class maintain insight into consequences for requirements to ensure that the final product possesses the key features needed to drive market demand. The Best-in-Class further support this effort by updating their test plan for verification and validation as changes are made.

Requirements Management

Managing system complexity starts with proper definition and management of requirements (Figure 5).

Figure 5: Requirements Management Capabilities



Source: Aberdeen Group, August 2012

Successful requirements management starts with an understanding of customer and market needs. Best-in-Class performers and their competitors recognize this and ensure that they identify customer needs before defining the requirements to respond to those needs.

The next step taken by the Best-in-Class addresses the delays integration problems pose to time-to-market (identified as a top challenge by 47% of study participants). As systems become more complex, there tends to be more interfaces which become even harder to management. The Best-in-Class are 80% more likely than their competitors to respond by thinking up front about the definition and applying consistency to those definitions. This makes the interfaces easier to manage, even as complexity grows.

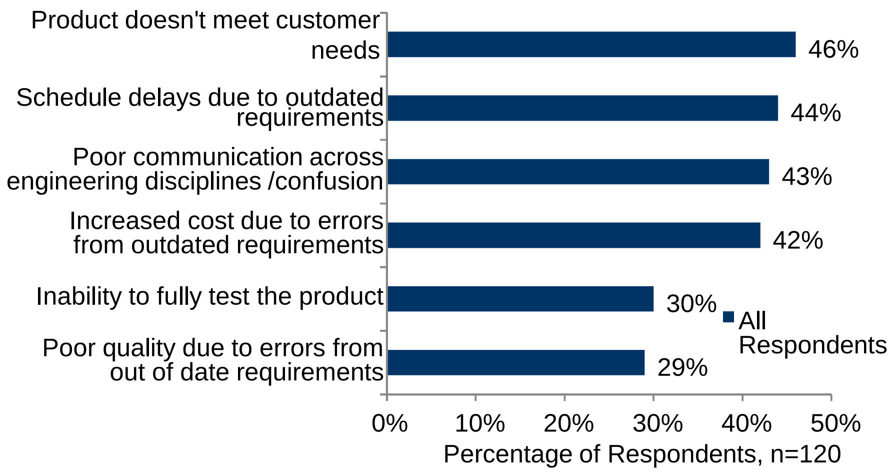
Increasing system complexity also makes it more difficult to assess system functions. The Best-in-Class simplify this process by using visual diagrams and models to define requirements. The Best-in-Class recognize that, in this respect, a picture is worth a thousand words. Using visual formats such as diagrams or models makes it easier to communicate and understand requirements. This simplifies the management of requirements as well.

Requirements Traceability

Lack of traceability adds to the difficulty of requirements management. Effective requirements traceability means creating links between requirements to ensure that dependencies are recognized so that the impact of changes can be understood.

Only 1% of survey respondents indicate that requirements traceability has no impact. This means that the vast majority of study respondents find that traceability represents an important element of requirements management. It should not be a surprise to learn that the consequences of poor traceability are often dire. The top reported impact is that the product does not meet customer needs (Figure 6). This is often a result of changes to requirements that are made without an understanding of customer needs. As a result, changes are made blindly to satisfy immediate pressures. While this is sometimes necessary, a missed customer need means a lost revenue opportunity.

Figure 6: Consequences of Lack of Requirements Traceability



Source: Aberdeen Group, August 2012

Respondents also reported that poor traceability often results in outdated requirements. When organizations lack insight into the requirements impacted by a change, those requirements become outdated, often without anyone noticing until it's too late. This causes schedule delays once it is discovered that requirements aren't met as well as time wasted unnecessarily to fix the resulting problems. These errors also result in increased costs in scrap and rework when problems are found. Alternatively, these problems create quality issues as other development work is done using the wrong requirements. Often, the problems manifest themselves in component interfaces. Finally, out-of-date requirements create confusion in validation, which makes it impossible to fully test the product. In the worst cases, this results in products that are released to the market with quality problems.

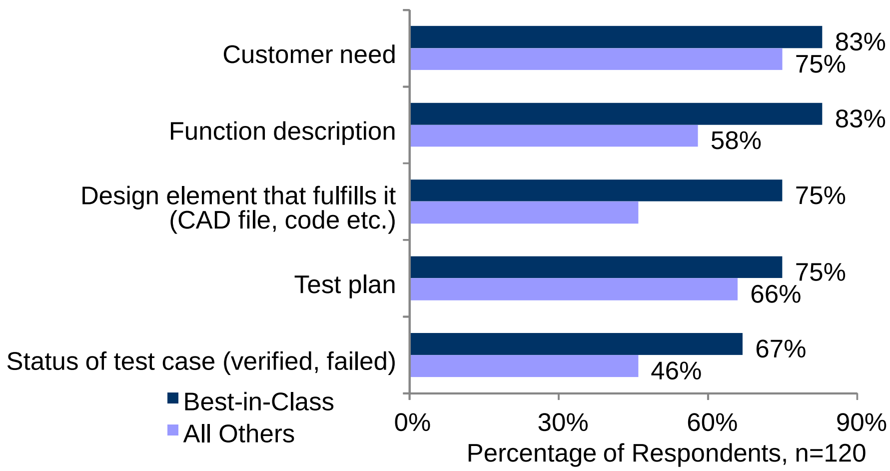
Effective traceability provides a mechanism for engineers across disciplines to understand the dependencies on the functions of their components. Without this visibility, it becomes harder to identify how changes to the

Telephone: 617 854 5200
Fax: 617 723 7897

component impact the rest of the subsystem, impeding communication and collaboration across engineering disciplines.

The Best-in-Class address this problem by linking requirements to key product information. These links make it easier to understand how a change to the requirements impacts customer needs. More comprehensive linking creates superior visibility into all system interactions. For example, the Best-in-Class are able to identify how a change in a CAD file impacts a requirement and then they can trace that change to the customer need (Figure 7).

Figure 7: Product Information Linked to Requirements



Source: Aberdeen Group, August 2012

Traceability is also needed between the requirement and test plan, so that the test plan can be updated as requirements change. This ensures that requirements validated are the current requirements. Additionally, when a test case fails, the Best-in-Class provide traceability back to requirements to understand the failure's impact. This in turn is traced to the different design elements, providing insight into the root cause of the failure.

Recommended Actions

As organizations incorporate embedded software components to get leg up on competitors, the amount of embedded code they generate has grown by 25%. The challenges of incorporating software, electronics, and mechanical components in a single product add considerable complexity, especially in the form of interdependent product systems and subsystems.

Organizations seeking to implement a Best-in-Class change process must:

- **Update change management processes across engineering disciplines**. This starts with a focus on requirements management and traceability so that it is easier to track down the full impact of a change, accommodate growing system complexity and ensure that key feature requirements will be met.

Telephone: 617 854 5200
Fax: 617 723 7897

- **Define requirements in a way that is easy to understand**. Visual format makes it easier to visualize what is expected of the system, even as it becomes more complex.
- **Bolster requirements traceability to ensure insight across complex systems**. This links customer needs to requirements, which are, in turn, linked to a design components. This also provides understanding of the full impact of a change and ensures anything done can be traced back to what the customer or market needs. It thus becomes easier to understand the potential impact on a product's market potential.

By ensuring that their development processes provide for continuing reliability and traceability, the Best-in-Class are able to more effectively incorporate the features their customers demand and launch more profitable products than their competitors.

For more information on this or other research topics, please visit www.aberdeen.com.

Related Research	
Embedded Software: The Future of Innovation in Tomorrow's Products; October 2011 *System Design: Get it Right the First Time*; August 2011	*The Secret of Designing Products Customers Love: Manage Requirements Effectively*; December 2009

Author: Michelle Boucher, Senior Research Analyst, Product Innovation and Engineering (michelle.boucher@aberdeen.com)