1-1-2009

Addressing business agility challenges with enterprise systems

D. Goodhue
University of Georgia

D. Chen
Texas Christian University

M. C. Boudreau
University of Georgia

Ashley R. Davis
San Jose State University, ashley.r.davis@sjsu.edu

MIS Quarterly Executive

Addressing Business Agility Challenges with Enterprise Systems[1]

Dale L. Goodhue
University of Georgia (U.S.)

Daniel Q. Chen
Texas Christian University (U.S.)

Marie Claude Boudreau
University of Georgia (U.S.)

Ashley Davis
San Jose State University (U.S.)

Justin D. Cochran
University of Pittsburgh (U.S.)

Executive Summary

It is clear that systems agility (i.e., having a responsive IT infrastructure that can be changed quickly to meet changing business needs) has become a critical component of organizational agility. However, skeptics continue to suggest that, despite the benefits enterprise system packages provide, they are constraining choices for firms faced with agility challenges. The reason for this skepticism is that the tight integration between different parts of the business that enables many enterprise systems' benefits also increases the systems' complexity, and this increased complexity, say the skeptics, increases the difficulty of changing systems when business needs change.

These persistent concerns motivated us to conduct a series of interviews with business and IT managers in 15 firms to identify how they addressed, in total, 57 different business agility challenges. Our analysis suggests that when the challenges involved an enterprise system, firms were able to address a high percentage of their challenges with four options that avoid the difficulties associated with changing the complex core system: capabilities already built-in to the package but not previously used, leveraging globally consistent integrated data already available, using "add-on" systems available on the market that easily interfaced with the existing enterprise system, and vendor provided "patches" that automatically updated the code. These findings have important implications for organizations with and without enterprise system architectures.

BUSINESS AGILITY CHALLENGES

- *All your business systems in Russia are denominated in dollars and the Russian government decides that, with three months notice, all transactions must be conducted in rubles.*
- *Oil is one of your main raw materials and the cost of oil suddenly starts fluctuating wildly. You realize that if you don't change how your systems deal with cost variations, profit will take a major hit and you'll be raising prices just as the cost of oil goes down.*

It is clear that "systems agility" (i.e., having a responsive IT infrastructure that can be changed to meet changing business needs) has become a critical component of organizational agility.[2] The above examples highlight how competitive environments require prompt responses to change both business processes and the information systems that support these processes.

What is not clear is whether the large integrated packaged systems referred to as enterprise systems[3] so prominent in today's businesses actually help or hinder systems agility. There is compelling evidence that the move toward enterprise systems over the

MISQE is Sponsored by

KELLEY SCHOOL OF BUSINESS
INDIANA UNIVERSITY

SIM
Delivering Business Value Through IT Leadership

1 Cynthia Beath is the accepting Senior Editor for this article.
2 For example, see Sambamurthy, V., Bharadwaj, A. and Grover, V. "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," *MIS Quarterly* (27:2), 2003. This conceptual article places information systems in the context of three key organizational capabilities that influence firm performance: agility, digital options and entrepreneurial action.
3 Enterprise systems, sometimes call enterprise resource planning (ERP) systems, are purchasable software packages that integrate processes and data across many business functions and geographies. The dominant players in the enterprise system market are currently SAP and Oracle.

last 10 years has had positive financial and operational impacts.[4]

There is also compelling evidence that enterprise systems allow firms to tailor their business processes to closely match the current business environment, thereby improving efficiency and effectiveness. However, some of the evidence of positive impacts has come from firms whose business processes are relatively stable,[5] and recent reports have continued to question whether the extra complexity of these highly integrated systems lowers a firm's capability to change its systems in response to business changes.

For example, in a 2007 *Sloan Management Review* article,[6] Rettig suggested that "Rather than agility, [enterprise systems] have produced rigidity and unexpected barriers to change (p. 25)," and a 2007 article in *The Economist*[7] quoted the old joke that "implementing SAP [the leading enterprise system] is like pouring concrete into a company." These articles suggest that whatever benefits enterprise systems may bring, they could be a bad choice for firms seeking corporate agility.

Some academics have also raised the same issue:

> *"In some respects, recent developments in [information and communication technology] such as enterprise systems have had a negative impact on agility and sustained competitiveness rather than the positive impact most often expounded in the mainstream literature. Companies that attempted to utilize [these technologies] to increase efficiency and reduce costs may have lost agility in the process."*[8]

These persistent concerns motivated us to conduct a series of interviews with business and IT managers in 15 firms competing in a variety of industries. Nine of the 15 firms were very large (>50,000 employees) and only two had less than 10,000 employees. We interviewed managers operating at the interface between the business and the IS group. These managers identified important systems agility challenges they had recently faced, and how they responded to them. Thirty-seven of the 57 challenges they identified involved in-place enterprise systems; 20 involved only non-enterprise systems. (See the Appendix for more details of our methodology; the names and identities of the firms are disguised.).

Overall, we found that rather than constraining agility, enterprise systems seem to *enable* systems agility. Before we turn to how these firms responded to their agility challenges, we first present the types of agility challenges they faced.

CATEGORIZING THE AGILITY CHALLENGES

The 57 challenges varied from the predictable to the totally unexpected, and from the need to interact differently with customers to the need for greater internal efficiencies. Figure 1 describes two agility challenges that were successfully met, one in an enterprise system application domain and one in a non-enterprise system domain. Figure 2 describes two agility challenges that were not successfully met, again, one from each type of system.

To draw out patterns and insights, we categorized the 57 systems agility challenges along several dimensions: what was the firm's relevant IT infrastructure (enterprise system or non-enterprise systems[9]); what type of *business* agility challenge needed to be addressed; whether a specific challenge might have been anticipated and built into the requirements for a prior system purchase or development initiative; and how the IS group did, or did not, meet the challenge (i.e., the "response" to the challenge).

We divided the 57 agility challenges into the five categories shown in Figure 3. The table shows the percentage of the challenges that fit into each category, together with the percentages of challenges in enterprise system and non-enterprise system domains accounted for by each category.

4 The following articles all present evidence of the positive impact of enterprise systems: Davenport, T. H., Harris, J. G. and Cantrell, S. "Getting more results from enterprise systems," in: *Strategic ERP: Extension and Use,* Bendoly, E. and Jacobs, F. R. (eds.), Stanford University Press, 2005, pp. 71-84; Anderson, M. C., Banker, R.D. and Ravindran, S. "The New Productivity Paradox," *Communications of the ACM* (46:3), 2003, pp. 91-94; Bendoly, E. and Kaefer, F. "Business technology complementarities: Impacts of the presence and strategic timing of ERP on B2B e-commerce technology efficiencies," *Omega* (32:5), 2004, pp. 395-405.

5 The following two articles give evidence of the positive impact of enterprise systems, but acknowledge that the domains studied are fairly static: McAfee, A. "The Impact of Enterprise Information Technology Adoption on Operational Performance: An Empirical Investigation," *Production & Operations Management* (11:1), 2002, p. 33-53; Davenport et al., 2005, Op cit.

6 Rettig, C. "The Trouble With Enterprise Software," *Sloan Management Review* (49:1), 2007.

7 "Liquid Concrete," *The Economist,* September 13, 2007.

8 Galliers R., "Strategizing for Agility: Confronting Information Systems Inflexibility in Dynamic Environments," in *Agile Information Systems*, DeSouza, K. (ed.), 2007, p. 4. See also Mooney, J. D. and Ganley D. "Enabling Strategic Agility Through Information Systems: The Roles of Loose Coupling and Web Services Oriented Architecture" in the same book.

9 It was quite easy to distinguish—interviewees had no difficulty telling us when an agility challenge involved an enterprise system.

Figure 1: Examples of Successfully Met Agility Challenges

	Enterprise System	Non-Enterprise System
Company	**"FoodDist"**	**"InsureCo"**
Business Context	FoodDist distributes its products all across the U.S.. An important expense item is the correct and rapid loading of its trucks. Operational efficiencies in this process go straight to the bottom line.	InsureCo has legacy systems for its policy management but before Hurricane Ivan in 2004, the systems supporting the claims adjusters had been mostly paper-based. After the hurricane, with the huge number of claims, it became clear that better systems were needed. InsureCo responded with homegrown standalone systems focused solely on claims and the needs of claims adjusters, not integrated with the other legacy systems.
Systems Agility Challenge	FoodDist wanted to track performance of its warehouse crews so it could give performance incentives. Unfortunately, the existing business process made this impossible—the enterprise system prepared delivery orders and the supervisors allocated these to warehouse crews as they saw fit, with no record of which crew got which order.	Adjusters asked for reports that group claims by county. Since county was never a field that had been captured before, this created a challenge.
Systems Agility Response	FoodDist combined a packaged third-party kiosk system to randomly assign delivery orders and record which crew received them, with pre-existing performance tracking capability in its enterprise system.	Since the claims system was not integrated with other systems, it was fairly easy to add a field for county to the data input screens and the database, and then to build reports that sorted claims by county.
Business Impact	This combination enabled FoodDist to track crew performance and implement incentives.	These changes met the needs of the adjusters. InsureCo is now planning to use a purchased add-on system or service that will take a collection of claims (by county) and calculate the best route for adjusters to follow in visiting all the claim sites.

Overall, 30% of the challenges were concerned with improving effectiveness (such as changing the way sales leads are allocated to sales channels) or efficiency (such as cutting costs by moving to global procurement or improving the product distribution process). Another 18% of the challenges involved responding to changes in government regulations (such as tax changes, or meeting new requirements imposed by the Sarbanes-Oxley bill or responding to new regulations about bio-tech products). 16% of the challenges were concerned with addressing customer requests for new features or addressing prospective new requirements to match or leapfrog competitors. Supporting the firm's growth (most commonly, the acquisition of a business unit) followed with 12% of the challenges. Included in the "Other" category are less common types of agility challenges, such as improving analytical capabilities, acquiring new data, and facilitating reorganizations.

Analyzing the challenges by enterprise system and non-enterprise system application domains exposes some interesting differences between the two. *Efficiency/effectiveness* challenges were more commonly mentioned by managers in enterprise system domains (38% vs. 15%). Our interpretation is that companies with enterprise systems may devote more management attention to looking for process improvements: they may have more of a business process mentality encouraged by the visibility into corporate-wide, integrated processes across business units. In contrast, the *Growth* category of agility challenges was mentioned more commonly in non-enterprise system domains (20% vs. 8%). We suspect that because enterprise systems are built to be scalable

Figure 2: Examples of Unsuccessfully Met Agility Challenges

	Enterprise System	Non-Enterprise System
Company	**"ServiceCo"**	**"GlobalProducts"**
Business Context	ServiceCo is a large global company selling a variety of products and services. One common type of order involves two different types of products and a service. Obviously, pricing for the combined order is quite different than it would be if the products and service were purchased separately. Although ServiceSo uses enterprise systems, the fulfilment systems for the three different types of sales are handled by three separate enterprise systems, because of the widely different processes and concerns that have to be addressed.	GlobalProducts sells its product worldwide through a network of sales representatives and also through the Web. It has a collection of legacy systems customized over many years to meet its unique requirements. Originally, pricing was fairly standard and could be handled with a simple order entry system. Over time, GlobalProducts added more versions of its products and has began to offer customers special prices based on customer type, order size, possible future purchases and other factors.
Systems Agility Challenge	Customers complained about having three different orders and three different invoices. For both customer convenience and internal efficiency, ServiceCo wanted to have a single fulfilment system that could handle pricing and fulfilment for the combined purchase.	The pricing logic used in the quote system and in the shipping/billing system are now not always in sync—partly because much more is known by the time the bill is prepared, but also because the two systems were developed separately and have slightly different logic appropriate to their initial focus. This creates problems and confusion for the customer.
Systems Agility Response	Because such a combination was never anticipated in the design of the three different enterprise fulfilment systems (each quite complex), ServiceCo has, as yet been unable to develop a combined system. Instead it has jury-rigged a process that takes a single order, breaks it into its component parts for the three separate systems, then recombines the relevant information to create a combined invoice.	Both systems work well independently, but the costs of rewriting one or both are prohibitive.
Business Impact	This complicated process does meet customer demands, but does not contribute to efficiency.	GlobalProducts is working to bring the two systems into sync, but admits that it will probably take years.

and to accommodate growth, some managers with an enterprise system may not have considered these needs to be agility challenges.

This raises an important question: with better planning, could some of the agility challenges identified by our interviewees (such as growth by acquisition) have been anticipated at the time the original systems were designed? If so, the needed capabilities could have been built into the original systems, at far less cost than adding them later.

The example of ShipRight Enterprises, which provides firms with packaging and shipping-related material, clarifies the distinction. At ShipRight, logistical efficiency is key—the product is low cost and high volume; demand varies widely according to production levels of the firm's customers. To improve its agility, the firm needed to develop better demand forecasts, based on historical data. Such forecasts had not been included in the original systems, and many might say that this first agility challenge could have and should have been anticipated. But even if ShipRight had carefully considered its possible future needs when it first designed its systems, it might well not have anticipated the types of information that it would eventually want to incorporate into its demand forecasts (a second agility challenge): sales representatives' soft data about when customers

Figure 3: Categories of Business Challenges Faced by Our Interviewees[10]

Challenge Category	Total (%)	Enterprise System	Non-Enterprise System
Efficiency/Effectiveness	17 (30%)	38%	15%
Regulations	10 (18%)	19%	15%
Customer Requirements	9 (16%)	14%	20%
Growth	7 (12%)	8%	20%
Other: New Data, Pricing, Reorganizations, Reporting, etc.	14 (24%)	22%	30%
Total	**57 (100%)**	**37 (100%)**	**20 (100%)**

were planning major plant maintenance, or special promotions, etc. In hindsight, these are all things that could affect customer production levels and therefore demand for ShipRight's product. We judge that ShipRight could have anticipated the need for better demand forecasts (one challenge), but was unlikely to have been able to anticipate the need to incorporate the additional information (a later, second challenge).

Many practitioner articles focus on the importance of more careful planning to align information systems with the business as a way of achieving agility.[11] Our subjective assessment of whether the 57 agility challenges in our sample could have been anticipated at the time when the managers' current systems were acquired (purchased or built) is shown in Figure 4. This analysis suggests that careful planning will only fully anticipate about a third of the cases where systems agility will be needed. In particular, specific types of agility challenges such as regulatory changes or customer requests for new features are likely to be especially difficult to anticipate. In other words, system agility requires being able to change systems after they have been built and used for some time.

RESPONSES USED TO MEET AGILITY CHALLENGES

We grouped the different responses used by the companies we interviewed to address their agility challenges into the seven categories shown in Figure 5, which also shows the relative overall use made of each response. We have ordered the responses based on their overall prominence. After briefly describing each one, we will look at the different response profiles presented by challenges in enterprise system versus non-enterprise system domains.

The overall dominant response for meeting agility challenges is to use capabilities already built into existing systems, even if not previously used (29% of the total). The second most used response was building on top of an existing data and process integration platform, used 19% of the time.

The third most prominent category, at 16%, was "Unable to Address." These are challenges that were recognized by the firms, but the firms were not able to successfully address the challenge for one reason or another.

The next two most commonly used responses, tied at 12% each, were customizing the existing code and using add-on systems. A major contributor to the motivation for this article is that customizing existing software has become exceedingly complex. (In Appendix B we delve into the reasons why this is so,

10 The differences between the circled values are statistically significant at the .05 level.

11 For an article that emphasizes planning as a key component of agility, see Plumber, D. "Measuring Your Agility Quotient: Are You Ready To Be An Agile Enterprise?" Gartner Application Integration and Web Services Summit, December 2005.

Figure 4: Assessment of Whether the Agility Challenges Could Have Been Anticipated[12]

Challenge Category	No	Yes	Total
Efficiency/Effectiveness	8	9	17
Regulations	9	1	10
Customer Requirements	9		9
Growth	3	4	7
Other: New Data, Pricing, Reorganizations, Reporting, etc.	9	5	14
Total	38	19	57

Figure 5: Use of Responses, Overall and by Domain Type[13]

Response Category	Total (%)	Enterprise System	Non-Enterprise System
Use Built-in Capabilities	17 (29%)	42%	5%
Build on Data and Process Integration	11 (19%)	30%	0%
Unable to Address	9 (16%)	3%	40%
Customize Existing Code	7 (12%)	5%	25%
Use Add-on Systems	7 (12%)	14%	10%
Develop via Skunk Works	4 (7%)	0%	20%
Acquire Vendor Patch	2 (4%)	7%	0%
Total	57 (100%)	37(100%)	20(100%)

and the extent to which we might expect differences in difficulty for enterprise systems versus non-enterprise systems.) Add-on systems involve a firm purchasing a software package that can be "plugged in" to an existing system to provide extra functionality, without requiring much or any change to the existing system. Today there are many vendors eager to provide such add-on packages.

The least used responses listed in Figure 5 are skunk works development (7%) and acquiring a software patch provided by a vendor (4%). Skunk works are "quick and dirty" information systems solutions that are developed by personnel reporting outside of the traditional IS department, or by a special group formed by the IS department, to avoid development standards and practices usually required to safeguard the reliability and security of corporate software when it would excessively delay the response.

12 The differences between the circled values are statistically significant at the .05 level.

13 The differences between the circled values are statistically significant at the .05 level.

USE OF RESPONSES IN THE TWO DIFFERENT DOMAINS

We now describe each of the response categories in more detail and provide illustrations of how they were being used to address agility challenges in the two different application domains. Four of the responses shown in Figure 5 (using built-in capabilities, building on integrated data and processes, using third-party add-on systems and vendor patches) accounted for about 93% of the agility solutions used in enterprise system domain. The four most used responses for non-enterprise system applications, accounting for 95% of the total, were unable to address, skunk works development, customizing existing code, and add-on systems.

Most Common Responses in Enterprise System Domains

1. Use Built-in Capabilities. Figure 5 shows that 42% of the agility challenges involving enterprise systems were met by taking advantage of a built-in capability (perhaps a purchasing module already in place but not used). Only 5% of the non-enterprise system challenges were met in this way.

Since many enterprise system products are mature ones, and their vendors seek to sell to a large and diverse market, their products often have built-in capabilities that clients might not initially use but choose to use in the future. "SpecialtyCo" is a case in point. After contracting out its logistics services for some time, this firm decided to bring it in-house. Since the enterprise system package it was already using had complete logistics functionality, this change only required turning on that functionality in the configuration tables.

Pre-existing capabilities built into enterprise systems can also obviously significantly reduce the time it takes a company to respond a specific agility challenge. These capabilities are not typically present in non-enterprise system domains.

2. Build on Data and Process Integration. This response was used to meet 30% of the agility challenges involving enterprise systems. It was not used in any of the challenges involving non-enterprise system applications, most likely because none had truly integrated processes and data.

A good example of this response comes from "AgriCo," which sells agricultural seed, among other products. Seed inventories are classified not just by plant species, but also by germination date and quality (the amount of weed and other plant seeds). It is critical that AgriCo price the seed so that it is sold prior to its germination date (after which it is useless). But prices must also be set to ensure the firm has enough seed to meet late-developing requests for high-quality, high-priced seed.

Previously, when a customer requested seed of a particular species, purity and germination date, it was difficult to access and interpret the non-standard data held by each warehouse, which meant the search for the best product was limited to local warehouses. After initially implementing its enterprise system, the business realized it had continent-wide integrated inventory data available, and decided to expand its data analytic capabilities to better match the right seed to the right application for the right customer at the right price, all within the shortest cycle time. Thus it was able to leapfrog its competitors to give better service to customers at the same time as maximizing its profits.

To accomplish what AgriCo did, a firm without an enterprise system (or without at least a common inventory database and common order processing system) would have had to define a common set of data across all warehouses, add new data to those warehouse systems lacking certain data, translate the data from each warehouse into the common form and load it into a common database.

3. Use Add-on Systems. This response was used by firms in both domains. It was the third most frequent response in enterprise system domains (14%) and was also used to address 10% of non-enterprise system challenges. As the installed base of packaged systems has grown, third-party vendors have responded to the large potential market for specialty add-on systems that can easily be interfaced with existing packaged systems, especially those with large market shares. Market-leading enterprise systems vendors in particular have also recognized the value of the third-party marketplace and have designed selected "user exits" into their products, making interfacing even easier. The resulting enterprise systems can be thought of as a "backbone" of tightly integrated processes that include lots of valuable interdependencies, and strategically placed user exits from the backbone. These exits allow processing to pass to a third-party specialized module that makes a specific change in the data and then returns the processing to the tightly integrated backbone.

An example of using an add-on system to meet an agility challenge is provided by "FoodDist,"

which decided to move to global sourcing for all its purchases. Its desired solution was to use the built-in purchasing functionality already available in its enterprise system (but not previously used). However, FoodDist recognized that the enterprise system's way of searching for the lowest cost and most appropriate vendor was not effective enough to meet its needs. Thus, it identified a third-party add-on system that was more effective in finding the best vendor for a particular need. This add-on was plugged into a user exit built into the standard enterprise system, and is now used just to determine which vendor to place the order with. Once a vendor is identified in the add-on system, processing returns to the standard enterprise system purchasing process. Plugging in the vendor-choice module to the enterprise system's user exit did not require any changes to the enterprise system, and any subsequent changes to the add-on module can be made without changing the enterprise system.[14]

4. Install a Vendor Patch. Enterprise system vendors regularly respond to certain agility challenges that many of their customers face, such as a change in the U.S. tax code. The vendors provide a self-contained program ("patch") to automatically change all the relevant software code in their enterprise system to accommodate the needed change. While only 7% of the challenges involving enterprise systems were met in this way, this solution makes responding to certain challenges quite straightforward. It is also likely that interviewees with enterprise systems may have under-stated the use of this response, because it is an expected type of vendor maintenance, requiring much less management attention. This response will likely not be available except for enterprise or other packaged systems with a large market share.

Most Common Responses in Non-Enterprise System Domains

1. Unable to Address. Sometimes firms must accept that they are not able to successfully meet an agility challenge as quickly as they would like. The most striking contrast in Figure 5 is that 40% of the agility challenges involving non-enterprise system domains were not met in a timely way, compared with just 3% in enterprise system domains.

Two examples of this were provided in Figure 2. Another interesting example occured at "GlobalProducts," which has extensive legacy systems supporting its many unique business processes in a highly competitive global industry. Because of the difficulty of changing its legacy software, GlobalProducts has adopted a twice-a-year "limited release" strategy for U.S. operations and a one-year "limited release" for global operations to allow its analysts and programmers to accommodate all the interdependencies between its legacy software systems before any change to any of the systems is released. However, this conscious strategy does constrain GlobalProducts' ability to respond to other business agility issues that arise between these releases. In the words of one IT manager:

> *"If you're in [one of the new Asian republics] and a competitor offers a one-day service and you have only a three-day service, and you have to wait typically a year and a half before you even get into the development cycle, you're at a pretty bad competitive advantage in that particular country."*

2. Customize Existing Code. Customizing existing code was the response used to meet 25% of the agility challenges in non-enterprise system domains, compared with only 5% in enterprise system domains. The IS community generally understands and accepts that customizing existing code is difficult and expensive. In Appendix B we provide background information for why this is true, and the extent to which we might expect enterprise systems to be more or less difficult to change than non-enterprise systems. Although many early adopters of enterprise system packages made extensive source code changes, today this is much less common.

3. Skunk Works Development. "Skunk works" refers to a approach for solving an urgent business problem that can't be solved quickly enough using an organization's normal IS development approach. Skunk works may use non-IS people or non-standard technologies and typically avoid the careful controls that IS groups insist on to ensure long-term quality and fit with existing systems. Thus, a skunk works development focuses on solving an immediate agility challenge, but may also require other follow-on modifications for long-term effectiveness. This response was used to address 20% of the challenges in non-enterprise system domains, but was not used for any of challenges involving enterprise systems.

An example of a skunk works development is provided by "FreeStyle," a worldwide service company. FreeStyle's top management was considering whether to move to a radically new product line, and desperately needed an automated

14 See discussion of the fourth or "modular" stage of IT architecture evolution in Ross, J. W., "Creating a Strategic IT Architecture Competency: Learning In Stages," *MIS Quarterly Executive* (2:1), 2003, pp. 31-43.

capability to capture new information about its existing customers from its salespeople. It was important to collect the new data in a form that could be analyzed quickly, and via existing business processes so salespeople wouldn't have to learn new processes. Management discussed the requirements with the IS group, but were told it would take at least six months—which would be too late. To speed things up, a computer-savvy group in marketing was tasked with rapidly jury-rigging a new Web-based system that was piggy-backed onto the existing Web-based production system.

The business captured the information they needed in time—and they were able to roll out the new product line. The downside is that when the new system went down (which happened frequently), it also brought down the whole production system. However, the new capability had so much business value that the decision was made to keep it in place, and after two years the IS group had "hardened" the new system to protect it from failure to the same extent as the other IS systems.

LESSONS FROM OUR RESEARCH

Enterprise Systems Need Not Constrain Agility

The primary lesson from our research is that IS managers and business executives have found that enterprise systems offer different options than non-enterprise systems to respond to the needs for business agility. The reality is that all of today's installed systems can be difficult to change: both enterprise and non-enterprise systems are quite a bit more complex than older "silo" systems because both incorporate many beneficial linkages between different parts of business functions. The advantage of enterprise systems is that they provide more options for responses to meet agility challenges that don't require changing the backbone of tightly integrated program code, and it is these options that so often helped the enterprise system companies in our study to be able to respond.

Changes to the Backbone Will Still Be Challenging

Of course, there is always the possibility that a particular business agility challenge might require changes to the actual backbone processing and logic—in other words, to the way in which the many functions of the business are linked. Making such changes will inevitably be difficult, whether the links are implemented through special programs that transfer data from one application to another (as in older "silo" legacy systems), or through a common database. Appendix B discusses why such changes are so difficult, and why we might expect different challenges from enterprise and non-enterprise system domains.

How often such backbone changes need to be made in any core system (enterprise or non-enterprise) will depend on the insight that was used when the core system was designed (either by vendors or by in-house development) and the way in which it allows specialized modules to be attached to it. The design effort invested by vendors to make their enterprise system backbones as future-proof as possible is probably greater and more effective than the design effort typically involved with non-enterprise systems that tend to be gradually designed and patched together.

Be Aware of Your History of Agility Challenges

A second general lesson is that all businesses—whether they have an enterprise system or not—should become more aware of the agility challenges they have faced in the past and the way in which they were met or not met. One way of doing this is to identify how and through what means business agility challenges are passed to the IS department, and to develop procedures to track them through to their resolutions. It will be important to note which responses led to a successful outcome and which challenges could not be addressed in a timely way. The cost of collecting this information will be small, but it could help IS leaders identify the kinds of agility challenges likely to be faced in the future, and how well the current application architecture is performing in meeting these kinds of challenges. In turn, this could result in identifying some adjustments or major actions that might improve systems agility. Such information could also be invaluable when making the case to business management for investments in a more flexible architecture.

Advice for Companies With and Without Enterprise Systems

Another set of lessons depends on the current infrastructure and future intentions of a particular

company.[15] We provide advice for three different situations:

- Companies that already have an enterprise system
- Companies considering moving to an enterprise system
- Companies that are not planning to replace their non-enterprise systems but are interested in achieving as much agility as possible. Such companies may have many unique processes or too much invested in their legacy systems to replace them with a packaged enterprise system.

Companies With Enterprise Systems. Our study shows that agility can be enabled by an enterprise system's built-in capabilities, its integrated data platform, and add-ons plugged in to its user exits. This would suggest that the IS group in a company with an enterprise system should ensure it:

1. Is aware not just of the enterprise system's capabilities already used, but those not yet used
2. Understands the placement and workings of user exits in the enterprise system
3. Is familiar with the range of add-on modules available on the market, or engages consultants who have that knowledge.

The IS group's knowledge in each of these three areas should then be assessed against the history of agility challenges the firm has already faced. This assessment may suggest better ways to address future challenges.

Finally, firms with enterprise systems should try to avoid creating complicated linkages outside of the systems. The best type of add-on system is one that exits the enterprise system, provides some useful additional processing, and then returns to the enterprise system at the point from which it started.

Companies Considering Enterprise Systems. There is a great deal of valuable literature and consulting advice covering the significant challenges of choosing and implementing an enterprise system. Based on our research, we offer the following additional advice:

1. As you evaluate candidate enterprise systems, pay careful attention to the extent of, placement of and ease of using user exits
2. Look carefully at the accessibility and ease of understanding the integrated data provided by the system, which could be the platform on which you build your next new business capability
3. Recognize that vendors with larger market shares will probably provide more functionality (even if much of it seemingly will not initially be needed) and that third-party vendors are more likely to build add-on modules for the products of these vendors.

Just as for companies that already have enterprise systems, assessing these three areas against the history of agility challenges already faced may help the IS group to focus better on what kinds of functionality, user exits and add-on packages it should be looking for to meet future challenges.

On the other side of the coin, buying an enterprise system with more functionality than you need saddles you with unnecessary operational complexity, and buying a system whose processes are too different from what you need will bring problems that are hard to fix.

Companies Not Planning to Replace Non-Enterprise Systems. Companies that decide to stick with their legacy non-enterprise systems can incrementally lessen the difficulty of agility challenges by working to evolve their systems toward an architecture that emulates an enterprise system architecture. Some specific actions that may or may not already have been considered are:

1. When modifying systems or implementing new systems, expand the scope of investment and increasingly build standardized systems for the firm's key business processes.
2. Consider how user exits might be incorporated into new or modified systems so that add-on packages could be used to address locally unique needs or future agility challenges. The history of recent agility challenges will help the IS group to focus better on what kinds of user exits and add-on packages it should be looking for to meet future challenges.
3. Where possible, move toward building global, or at least consolidated, databases of standardized data.
4. Work toward having standardized "backbone processing" for core processes, with unique or nonstandard needs provided by modules connected to, but not disrupting the backbone.

15 We note that this advice is quite consistent with that suggested in Ross, J. W., 2003, op. cit.

Unlike firms with enterprise systems, it is much less likely that these firms will have currently unused functionalities in their existing systems to respond to new business challenges. However, they can invest in the development of an infrastructure of integrated data and user exits—which were the second and third most commonly used responses for addressing agility challenges in firms with enterprise systems. The aim here would be to provide the firm with additional options beyond skunk works and expensive customization of code.

CONCLUSION

Although our research is based on interviews with a small number of companies, it provides evidence that contradicts the persistent views in the literature that enterprise systems are more likely to constrain companies' abilities to respond to business agility challenges. Although the user exits and add-on possibilities of enterprise systems also sometimes raise competitive concerns about system uniformity because of the use of the same "backbone" business processes, the larger the market share of an enterprise system vendor, and the more standardized its user exit interfaces, the more third-party vendors will emerge to provide special-purpose add-ons. The availability of these add-ons creates a wide range of different capabilities that firms can attach to their backbones to meet their unique needs and to respond to agility challenges.

IS leaders who want to create an architecture that can enable systems agility, with or without enterprise system packages, should think in terms of backbones that tightly integrate processes and produce integrated data, and provide appropriate user exits for specialized, modular add-on capabilities. That, we argue, is the way to achieve corporate systems agility.

APPENDIX A

RESEARCH METHODOLOGY

The 15 firms that participated in the research were selected opportunistically through available contacts between businesses and MIS departments at the University of Georgia and Texas Christian University. The industries and size of the firms (number of employees) is shown in the Table below, using fictitious names. We worked with our contacts to identify appropriate individuals who were close to the interface between IS and the business (reporting into either the IS or the business side), and at a sufficiently high level of the firm to understand a variety of business agility challenges and how they were resolved.

We conducted semi-structured interviews lasting about an hour. About half the interviews were taped and transcribed; the other half were summarized by two reviewers within a day of the interview, and the write-ups were sent to the interviewee for verification

Profile of the 15 Firms Involved in the Research

Name (Fictitious)	Industry	Employees
ServiceCo	Computers/Software/Services	Over 50,000
SoftwareCo	Computers/Software/Services	Under 10,000
HighTech	Consumer Products	10,000-50,000
CountryWide	Consumer Products	Over 50,000
FoodDist	Food	Over 50,000
SpecialtyCo	Industrial Products	10,000-50,000
AgriCo	Industrial Products	10,000-50,000
InsureCo	Insurance/Banking/Brokerage	Over 50,000
OldLine	Insurance/Banking/Brokerage	10,000-50,000
Assurance	Insurance/Banking/Brokerage	Over 50,000
FreeStyle	Transportation Related	Over 50,000
GlobalProducts	Transportation Related	Over 50,000
ShipRight	Transportation Related	Under 10,000
UtilCo	Utilities	Over 50,000
BestUtil	Utilities	Over 50,000

or correction. Coding of the agility challenges was done independently by multiple researchers, with comparison and adjustment after the original coding.

APPENDIX B

Why Business Application Software is Hard to Change

What is it that prevents IS organizations from effortlessly adjusting their current IT infrastructures to new forms that support new business requirements? The answer is that application software and data must mirror business processes, and business processes are often very complex. For example, business process rules might specify how many different pricing schemes there are and how many types of discounts are possible, and the factors to use in determining the pricing scheme and type of discount for a particular order. They may also specify how to decide which plant to use as the source of the product, or how to determine whose order gets filled when there are not enough units of the product to go around. All these general and special business process rules must be made explicit in the software.

Increasingly, businesses want and need the efficiency, effectiveness and end-to-end operational responsiveness that come with tight integration across geographies and functional areas. This cannot happen without "linking" different parts of the business through automatic flows of data. These data flows can be implemented in different ways. In non-enterprise system domains, special programs periodically copy data from one relatively standalone functional system and send it to another. In enterprise systems, different functional areas of the system all use a common database. Either way, linking functions together increases the total complexity of the information systems.

Stated differently, information systems are complex because, in today's world, businesses are complex. Whether they are non-enterprise systems with data flowing from one system to another, or enterprise systems with common databases, the complexity created by integrating across functions and organizational divisions makes systems harder to change.

For any system, changing existing software to meet new requirements involves a three-stage complex human task:[16]

1. Understanding the *existing* business rules and how they are implemented in the *existing* software
2. Understanding the modifications needed (what new things the software must do)
3. Understanding the steps to take to convert the existing software into software that includes the new requirements.

All this needs to take place without damaging everything the software used to do.

A common view is that non-enterprise systems are less complex to change than enterprise systems because they are viewed as being relatively self-contained "silo" systems with limited interfaces to other systems, and therefore lower complexity. This view is represented in Figure B1, which shows three systems (a sales system, an inventory system and a production system) and the interdependencies between the them and their subsystems.

When first built, legacy systems often looked very much like those in Figure B1, with very few interdependencies between systems and subsystems. The problem is that, over time, business needs changed, and legacy systems had to be adjusted to meet new demands. For example, production planning might have to be based not just on shipments out of warehouses, but also on current sales, and even on a percentage of active sales leads that had been shown to predict next month's actual sales. Each new requirement created new interdependencies between previously separate systems, and needed a new small program that had to be run periodically to move data from one system to the next.

Each new interdependency did bring real business value, but it also increased the complexity of the legacy system. The result is these systems today look more like Figure B2 than Figure B1. Every one of the interdependencies shown in Figure B2 supports an important business need, but together they make it much more difficult to change. As one IT manager in GlobalProducts (which has non-enterprise, legacy systems) put it:

16 For a discussion of the human task of software maintenance, see Shaft, T. M. and Vessey, I. "The Role of Cognitive Fit in the Relationship Between Software Comprehension and Modification," *MIS Quarterly* (30:1), 2006, pp 29-55.

Figure B1: "Silo" Legacy Systems Originally Had Relatively Few Inter-System (and Subsystem) Links

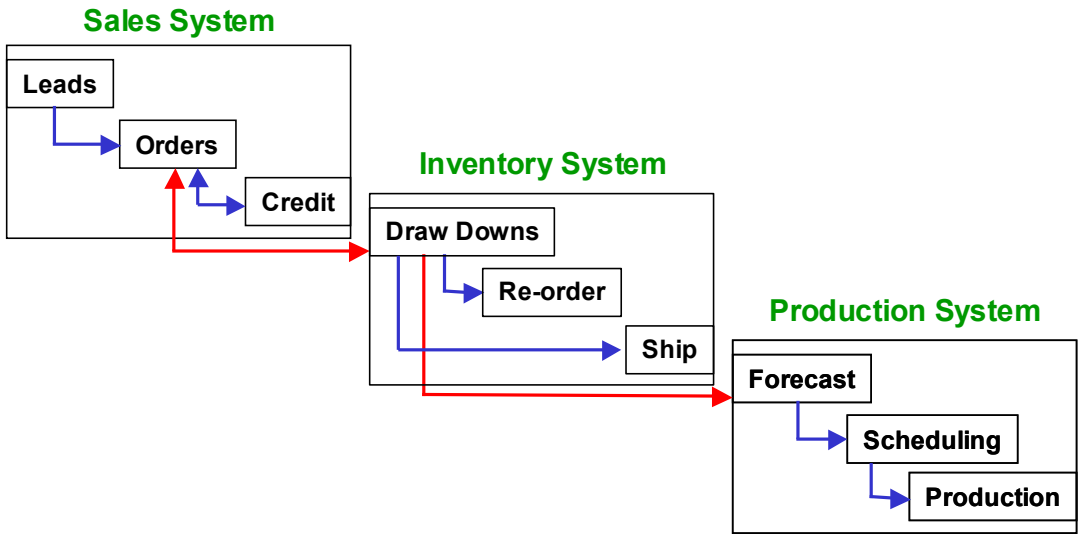
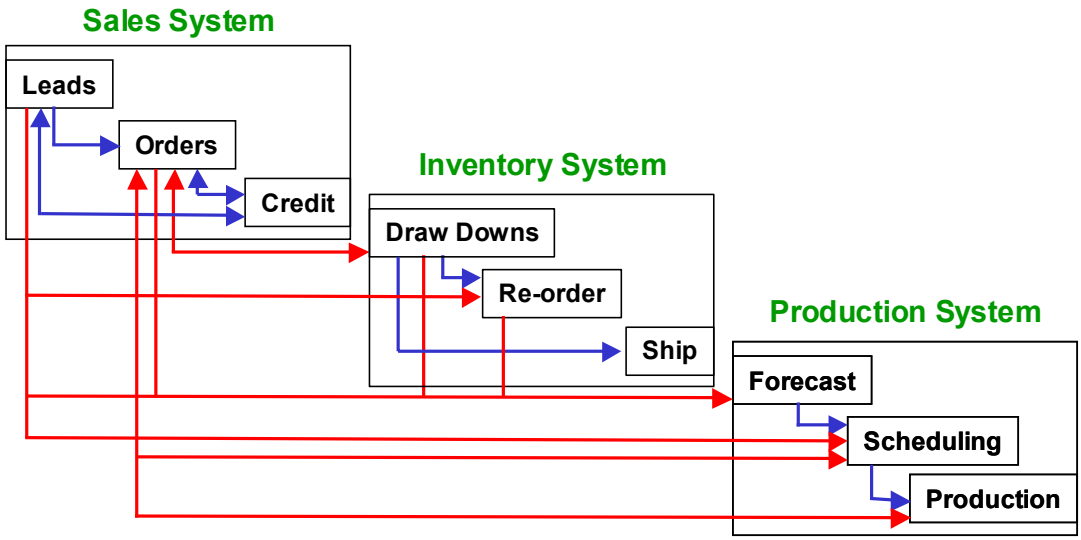


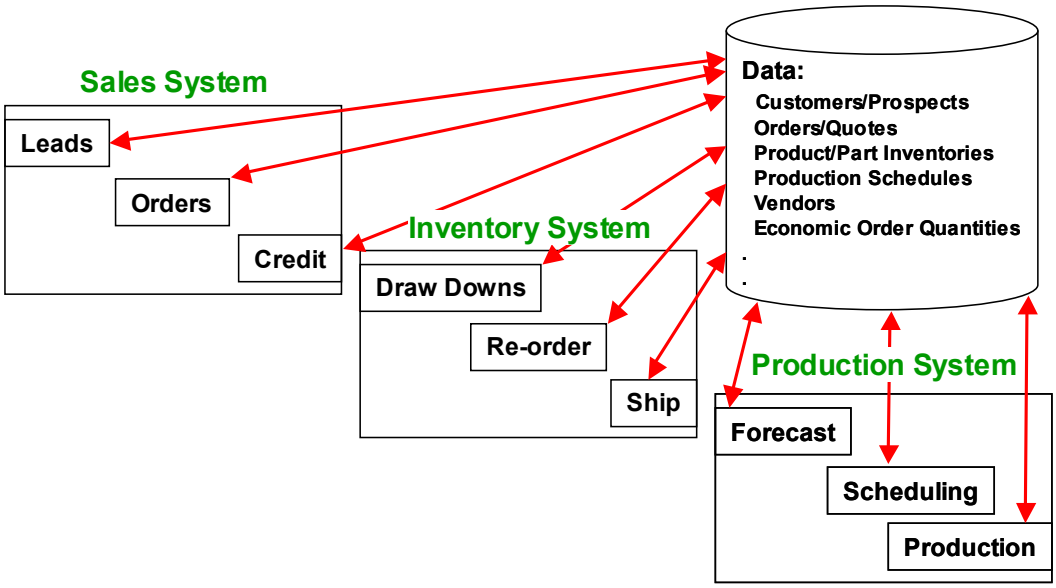
Figure B2: "Siloed" Legacy Systems Have Become More Complex with Lots of System and Subsystem Interdependencies



> *"Every time we come out with a new service or a new feature, it literally has a domino effect to, not only the customer-facing systems, but to the billing systems, the operational systems, even the payroll systems."*

The reality is that enterprise systems support at least as many links between different parts of the business as non-enterprise systems, but they implement the links differently. In enterprise systems, all subsystems are linked via a single integrated database, as shown in Figure B3. This means fewer lines on a diagram of links between subsystems, but the linkages incorporate all of the interdependencies shown in Figure B2, and each interdependency adds about as much complexity to the system as a whole.

Figure B3: Interdependence Links in an Enterprise System



One of the big benefits of enterprise systems is that they are explicitly designed to take into account interdependencies between different parts of the business. The result is that they generally do tend to support more integration between different parts of the business and to implement more (generally advantageous) interdependencies than even the most patched up of non-enterprise systems. Thus, when a change needs to be made to one part of an enterprise system, there is greater potential for impacts in other parts of the system.

But the issue of whether enterprise systems are easier to change than non-enterprise systems also depends on whether a local or a global change is needed. Consider a company that recently moved from country-specific order fulfillment information systems to a global enterprise system. The new enterprise system had to be modified in response to a business request from the company's Canadian unit to create a new option for placing orders and invoicing. When asked if it was harder to make the change in the new global system than it would have been in the old country-specific systems, an IS manager's response was unequivocal—it was harder with the enterprise system:

> *"There's no comparison in terms of complexity .... if you've got 70 or 80 countries on the model, there's a lot of room for problems afterwards because you're modifying something to suit the needs of a small number of countries at one point and then potentially causing issues with other processes."*

But consider the situation if this company had kept its 70 or 80 separate country-specific systems and was suddenly faced with an agility challenge that required a *global* change to the order-placement process in all the country units. In this situation, changing one very complex enterprise system does not seem so daunting, in comparison with changing 70 or 80 separate country-specific order processing systems, each different and each moderately complex. In general, local changes may be easier to make in non-enterprise systems, but global changes are generally easier to implement with enterprise systems.

ABOUT THE AUTHORS

Dale L. Goodhue

Dale Goodhue (dgoodhue@terry.uga.edu) is the C. Herman and Mary Virginia Terry Chair of Business Administration at the Management Information Systems Department of Terry College of Business, University of Georgia. He has published in *Sloan Management Review*, *MIS Quarterly Executive*, *Management Science*, *MIS Quarterly*, *Information Systems Research*, *Decision Sciences* and other journals. His research interests include measuring the impact of information systems, the impact of task-technology fit on individual performance, the management of data and other IS infrastructures/

resources, and the impacts of enterprise systems on organizations.

Daniel Q. Chen

Daniel Chen (d.chen@tcu.edu) is an Assistant Professor at the Department of Information Systems and Supply Chain Management, M. J. Neeley School of Business, Texas Christian University, Fort Worth, Texas. He received his Ph.D. in MIS from the University of Georgia. His research has been published in *Decision Sciences*, *MIS Quarterly Executive* and *Business Intelligence Journal*. His research interests include the organizational impact of IT infrastructures, IS leadership and IS strategic management.

Marie Claude Boudreau

Marie Claude Boudreau (mcboudre@terry.uga.edu) is an Associate Professor at the Management Information Systems Department of Terry College of Business, University of Georgia. She has researched the organizational change induced by information technologies and information systems, such as integrated software packages and open source software. More recently, she has been investigating the impact of information systems on organizational culture, ecological sustainability ("Green IT/IS") and system agility. Dr. Boudreau's research has been published in journals such as *Organization Science, Information Systems Research, MIS Quarterly, Journal of Management Information Systems* and others.

Ashley Davis

Ashley Davis (ashley@davissolutions.com) is an Assistant Professor at the Department of Accounting and Finance, San Jose State University, California. Her many years of ERP-related consulting experience, have driven her research interests on the impact of information systems on organizational agility, the impact of social capital on organizations, the impact of customization of large systems on the organization and open source systems. Her research has been published in *Communications of the AIS*.

Justin D. Cochran

Justin Cochran (jdcochran@katz.pitt.edu) is a Visiting Professor (PhD) at the Management Information Systems Department, Katz Graduate School of Business, University of Pittsburgh. His research interests include understanding the impacts of information systems on organizational agility and the factors affecting technology selection decisions.