

**Six Questions  
Executive Teams  
Should be Asking  
About Their  
Information Systems**

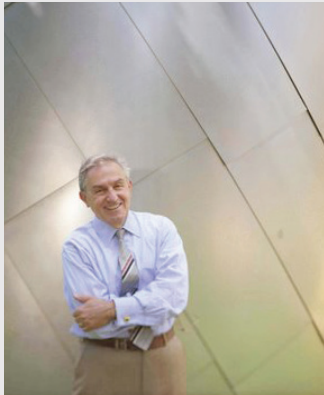


A PERSPECTIVE BY JIM CHAMPY, COLBY THAMES,  
JAY GRIEVES AND ROBERT MORISON



– Perspective by Jim Champy, Vice Chairman

**Time To Reengineer IT – Again** When Mike Hammer and I published *Reengineering the Corporation* in 1992, we saw the world of business frozen in complexity and outdated business processes. Work that seemed simple was taking too long to do and costing too much. Mike’s favorite expression was “don’t automate, obliterate.” We didn’t want companies just to automate old business processes. We wanted companies to focus on rethinking work processes first, then apply technology.



The role of information technology is dramatically different today from what it was in 1992. The Internet, the cloud, and mobility all conspire to make information technology the great enabler of dramatic business process change. Yet companies seem strangely frozen once again – not in old processes, but

in technology itself. The investments that companies have made in enterprise-wide systems have created efficiencies and helped standardized processes, but the scale and complexity of these systems now make both technological and operational changes difficult to do.

It’s time for companies to step back and radically rethink how systems get built and how information technology can enable process change. Companies need to be increasingly nimble and responsive to customer and market needs. No customer wants to hear that it will take a year or two for enhanced service. And no company executive wants to hear that an application is going to cost millions when consumers are buying apps for a few dollars apiece.

But the answer to reengineering IT is not simply to go to the Apple Store. It will take a combination of using new technologies and applying new development processes – processes that leverage the vast amount of information that companies now have available.

# Six Questions

With the astounding pace of technological innovation, we tend to assume that information technology stays far ahead of businesses' ability to use it well. That's often true of individual technologies. But businesses sometimes have to wait for technology to catch up – wait for the technological *combinations* needed to accomplish things they've been wanting to do for a long time. We're at such a point of technological convergence – we have the tools and techniques to reengineer how we build, manage, and capitalize on large information systems. In the accompanying perspective, Jim Champy makes the case that it's time to unleash the next wave of technology-based business process innovation. As an initial step, companies must reexamine the condition, performance, and roles of their major information systems. Here are six interrelated questions that CEOs, CIOs, and executive teams should be asking.

## 1. Do our major applications support our growth strategy?

Corporate enterprise resource planning (ERP) and other large enterprise systems enable one form of business growth – doing more of the same. As business volume expands, these applications can usually scale up. They can also support the biggest growth moves – mergers and acquisitions – if the company is quick and disciplined in making the enterprise systems standard across the enlarged operation. Most often, however, integration is slow and contentious. Competing sets of enterprise systems hinder coordination and operational performance, and they dilute the financial gains of consolidation.

But there's another form of business growth – through innovation. Here enterprise systems impede growth because it's hard to integrate innovative applications and processes with them, and because they lock an organization into one way of working and thinking.

*What's at the heart of your company's growth strategy – innovation, expanding along familiar lines, or both?*

## 2. Can we leverage existing applications assets in business experiments and innovations?

In a corporation equipped to innovate rapidly and dramatically, all of its information and technology assets would be available for experiments, pilots, and then implementing new ways of operating. All of its electronic data would be available for analysis. All of the business logic and methods represented in its software would be available to modify and assemble in new ways to test and implement innovations.

That ideal of modularity, connectability, and reconfigurability is old news to IT professionals. But very few companies come close to the ideal, in large part because their major systems represent a commitment to the opposite approach – closed, proprietary, tightly integrated. That's the case not only with ERP and other large software packages, but also with most home-grown enterprise systems. Very few large applications have been built for flexibility. Rather, these systems are brittle yet critical, to be maintained and not disturbed.

*If innovation is high on your business agenda, do you have the applications and technology architecture to enable it?*

### 3. Do our applications deliver the data we need for analytics and decision making?

One of the most common and longstanding complaints about enterprise systems is that they make it difficult to get at the information needed for management. This is ironic because so much effort goes into, for example, standardizing data as part of ERP implementation. But here's what happens: so much effort goes into standardization and installation that there's little *anticipation* of what information is needed for management, or how to manage differently with the information in hand. When new information needs arise or are recognized later, they're difficult to meet because the applications are focused on using predefined data for predefined purposes.

More and more companies are competing on analytics, differentiating themselves through their ability to generate new data and to make faster and better informed decisions, especially at the points of customer contact. Enterprise systems are major and essential data sources, but too often that data is difficult both to extract and to combine with other sources. Hence all the effort going into maintaining data warehouses apart from those applications, together with the interfaces that manage data feeds.

*In what ways could your business compete more effectively if it had more complete and available data?*

### 4. Can we take advantage of mobile applications across the enterprise?

Mobility is the rule, not the exception, these days. More employees spend more time away from the traditional workplace, both on the road and working from home. Even when in the traditional workplace, employees prefer to use their mobile devices – smartphones and tablets – to do things they used to do on laptops and desktops. And customers expect to connect with businesses via mobile devices, with their expectations conditioned by their experience with very slick consumer applications and interfaces.

The rapid rise of mobility puts enormous pressure on a company's entire information systems infrastructure. Remote employees need secure access to corporate applications, and it's not enough to try to "fit" everything on the smaller screens. Applications and interfaces need both to be streamlined for mobile productivity and to be expanded to take advantage of mobile capabilities such as location awareness. And given the pace of technological advance, mobile applications must be able to change quickly. Older systems, of course, were not built for mobile. Opening them to mobility is proving very difficult, and enterprise system vendors aren't doing much to help the cause.

*If your marketers identify a mobile application that will really drive sales, how fast can you follow through?*

## 5. Do our applications and our applications development capabilities differentiate us in the marketplace?

With software packages, you get standard applications for standard processes, which suffice for most everyday business activities. With some packages you get reasonably current best-of-breed applications, but with others you get what a CIO described to us as “someone else’s outdated process design.” With some packages you have many options and settings to choose from, but with all packages you have access to the same technology that everyone else does. Thus, differentiation only comes from how well you use the software operationally, how much you capitalize on the information generated, or how effectively you “surround” the systems with distinguishing capabilities. The last two are very difficult to do, given the lack of modularity and accessibility of most large applications, whether packaged or developed in-house.

The conventional wisdom of “buy, don’t build” carries serious implications for IT skills. With heavy reliance on packaged applications, IT organizations can see their systems development capabilities decline, even if headcount hasn’t been reduced. It’s common to find a majority of “developers” maintaining older home-grown systems and the complex of interfaces to the ERP, while very few are current and practiced in agile development techniques. Yet talented developers are key to business differentiation through IT.

*Can your IT organization deliver new capability at the pace of business need and opportunity?*

## 6. Are we getting our money’s worth from ERP and other enterprise systems?

The “bottom line” question can be trickiest to answer. Companies spend tens and sometimes hundreds of millions of dollars on ERP implementation, and all enterprise systems are significant investments. It takes a long time to recoup the initial investment, and in the meantime the ongoing maintenance and upgrade fees, and the large cost of maintaining interfaces, continue to mount. Once these systems are in place, it can be prohibitively expensive to replace them in whole or in part, and “sunk cost” thinking can impede even necessary change.

All that said, if your enterprise systems have injected consistency and discipline into major business processes and raised their performance, they may well have paid off in terms of operational gains. However, what are the opportunity costs of keeping inflexible applications in place? The costs of curtailed innovation and growth, of less-than-fully-informed management, of failing to capitalize on the productivity of mobility? Such opportunity costs cannot be calculated precisely, but they should be recognized and discussed.

*Can you develop an investment and action plan for a more agile computing platform?*

In most companies, the answers to most of these questions are “no.” But the answers can be “yes” – not all at once, but steadily and soon. It requires a different set of goals for individual applications and the computing environment as a whole. It requires fresh perspective, tools, and methods for systems development and management. It requires different relationships with vendors – and taking back control over technology assets. It requires daring and commitment to do things differently.

In the rest of this paper, we detail ways to get started – or accelerate your progress – revitalizing your information systems to keep pace with change and amplify business value.

## How Things Should Work

We recommend revisiting the goals of your major information systems and your computing environment as a whole. Consider these five basic objectives:

- **Data is available.** Businesses are forever finding new uses for existing data and useful opportunities to combine data in new ways. It's impossible to anticipate all these uses when applications and databases are designed or implemented. And it's impossible to anticipate what new data from outside the corporation will materialize and prove important. A company should be adept at gathering data from multiple sources and rapidly formatting it for analytics and decision-making. That includes extracting any and all data from legacy systems without disturbing them. Any data worth storing for one application is potentially worth exposing to others.
- **Business logic is available.** Similarly, the experience and logic, rules and procedures, coded into one application should be available to others. Making code available for reuse accelerates applications development because developers can modify and assemble components rather than coding from scratch. Reusing business logic has traditionally been even more problematic than making data available. The technologies for managing libraries of business rules and modules of code are in less widespread and effective use than those for data warehousing. Software vendors often don't make the source code available, and the code is difficult to extract and work with when they do.
- **Interfacing is easy.** There are conventionally two basic types of interfaces, the bridges between applications and the bridges to user devices. The first are troublesome when applications have different protocols, and especially when vendors release upgrades to applications software and all the interfaces to local

functionality must be revised. The second are troublesome as device types proliferate, and especially with the recent surge in mobile devices. The ideal would be an underlying architecture and set of standards that automate the development and management of all interfaces, inter-application and user interfaces alike. That's the only way to keep pace with and capitalize on the consumerization of IT.

- **Change is easy.** This is the paramount goal. If data and business logic are readily available, interface management is automated, and developers use agile methods, then everyday business changes can be incorporated into information systems in record time. Conscientious employees are continuously devising business improvements, but most are implemented as workarounds because it takes too long to get them into systems. The time between a front-line employee's good idea for a straightforward business improvement and implementation of that change should be at most a few weeks. Imagine the productivity gains of a true continuous improvement enterprise.
- **Experimentation is easy.** Large and potentially disruptive business changes must follow a different path: experiment with new designs in a "sandbox" apart from everyday business operations, pilot promising designs in realistic settings, scale up successful pilots, and eventually integrate into everyday operations. However, those same core assets – data, code, interfaces – should be available to enable and accelerate business experiments. With relevant assets available in a scalable cloud environment, the range of designs widens, pilots can be realistic, and big change can happen faster.

These goals are familiar to IT leaders, and they make intuitive sense to business leaders, including those with only passing familiarity with IT. Progressive organizations have been pursuing these goals for many years – but through a series of half

measures and fragmented efforts, and with little help from technology vendors. Progress was limited because the technology wasn't ready, and ambition was stifled because revamping the architecture and transforming the approach to applications management seemed too big and costly an effort. Buying inflexible software to meet most of the immediate need was always the path of least resistance.

## New Approaches, New Attitudes

The game has changed. Technology has moved from barrier to enabler. The old trade-offs among scale, speed, and cost are evaporating. We can store, process, and transmit almost unimaginable amounts of data at extraordinarily low cost. The modular and connectable architectural models of virtualization, cloud computing, and the Internet itself are prevailing. Advanced middleware is automating interfaces, thus enabling access to wide-ranging resources through standard browser portals. Advances in distributed security move us to anytime/anyplace computing on any commonplace device. Social media and other technologies of collaboration are transforming business processes, including how we develop and manage information systems.

Step back, look at those changes together, and you see that IT is working in a very different world from just a few years ago. It is now possible to:

- **Take a platform approach.** A *platform* is a set of assets whose roles and connections are defined so that they are all available to one another and can be configured in a variety of useful ways. Rather than being dedicated to one way of doing things, a platform is literally designed for change. But that doesn't mean sacrificing efficiency. One of the ways the technology assets are configured is to support today's business operations.

And because local improvements and fixes are so much easier to accomplish, a flexible applications platform can out-perform dedicated systems tuned to work just one way.

- **Take a migratory approach.** The platform approach is not a one-time, wholesale redesign of the technology architecture, but the problems are largely perceptual. Many IT people still think in terms of big projects to implement big designs. That's why so many large "service-oriented architecture" projects fail – they try to implement too much of a platform at once. The platform approach is not a one-time, wholesale redesign of the technology architecture, but rather a new set of design, development, and operating principles to be followed henceforth. Middleware, cloud, and other technologies enable you to migrate – aggressively if you like – by incorporating selected applications and other assets as you go. That, of course, lowers both cost and complexity barriers to action.

So technology is not the barrier. Systems development and management methods are not the barrier. The skills to exploit newer technologies and methods may be a barrier to progress in many companies. However, the biggest barriers are long-held attitudes and assumptions (some of them unstated) regarding the endeavor we call IT. Those attitudes stifle both understanding and ambition about what businesses can accomplish with technology, starting with the assets already in place. Here are three progressive attitudes that business and IT people alike should embrace:

- **Things change.** And we don't know what all the changes will be, no matter what our powers of prognostication. That's obvious and inevitable, and yet most information systems assume the opposite. Information systems cannot merely automate today's ways of operating – they must also keep tomorrow's options open. If they are not flexible and changeable, they paint us into corners, and all hell breaks

**Core and Edge** Over the past few years, we have witnessed an unprecedented stampede of information technology innovation. The confluence of social media, mobility, cloud computing, software as a service, and others is radically redefining the way that information technology drives and supports enterprises in every industry. Corporations must assess the uses and value of these tools.

A useful prism for viewing this issue is the core/edge dynamic and what it means for the way organizations innovate, deploy new information technology, and govern their IT assets.

- By *Core* we mean the existing operations, processes, systems, technologies, governance model, and mindset that run the enterprise.
- By *Edge* we mean any new process, tool, technology, or system that brings beneficial change or innovation into a company's operations.

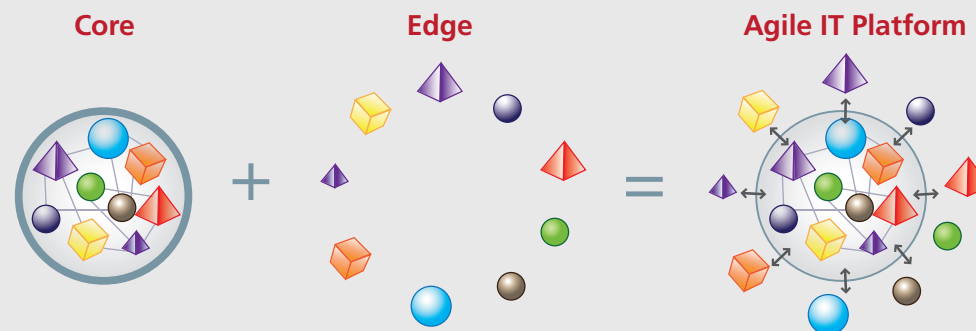
Growth through innovation is high on many a corporate agenda. But it's extremely difficult to introduce innovation directly into the core, with its focus on managing scale, driving efficiencies, making incremental improvements, and ultimately lowering the cost of business operations. Innovation has to happen at the edge of the enterprise, where new ideas incubate and the business experiments to discover what's possible. Too often core and edge are in conflict,

both technologically and organizationally. The core is called unimaginative, the edge rogue. But they must work together.

Where does the edge get its resources? Experimenting happens in a standalone (often cloud-based) computing environment, but data and sometimes business logic need to be borrowed from the core. And what happens with successful innovations? They need to be scaled up and integrated into the core, or else run alongside it and become the new core.

Thus, core and edge are in a continual and sometimes intricate dance. What kind of computing platform serves not just core activities, and not just edge activities, but the dance itself – the interplay between core and edge? It's a platform in which data and other assets can be extracted from the core without disruption, a platform that easily incorporates new technological components, and a platform that can put components working together in loosely coupled fashion. In other words, it's a highly flexible platform configured on the principles of modularity, connectivity, and scalability.

How can your organization harness that stampede of technology innovation to operate better, to innovate and then operationalize innovations, and to configure a truly flexible information systems and technology platform?





loose when business change forces systems change. A flexible platform keeps business options open and enables rather than impedes strategic change.

- **Learn as we go.** We don't know how things will change, and we don't know what solution works best until we try perhaps several. Yet traditional applications development methods assume the opposite: business people "specify" what they want, then IT people build to those specifications. The problem is that people often don't know what they want or what the alternative possibilities might be until they try doing something differently. It's far more important for business capability to be deployed quickly and modified easily than that it be specified completely up front. The new formula: build based on what you know, learn and improve through experience, and drive down the cost of modification.
- **Fit the system to the person.** Despite all the effort that goes into having users help specify systems, at the end of the day the users have had to adapt to whatever the technologists deliver. Business people need training in new workflows – that makes sense. But it's wasteful to train people to use complex and unintuitive interfaces because the specifications included too many bells and whistles. This matter is coming to a head with mobility and the consumerization of IT. Many employees are willing to bend or even break policy in order to use the devices, interfaces, and applications they like. You maximize productivity with applications flexible enough to handle workflow variations, plus interfaces flexible enough for users to customize.

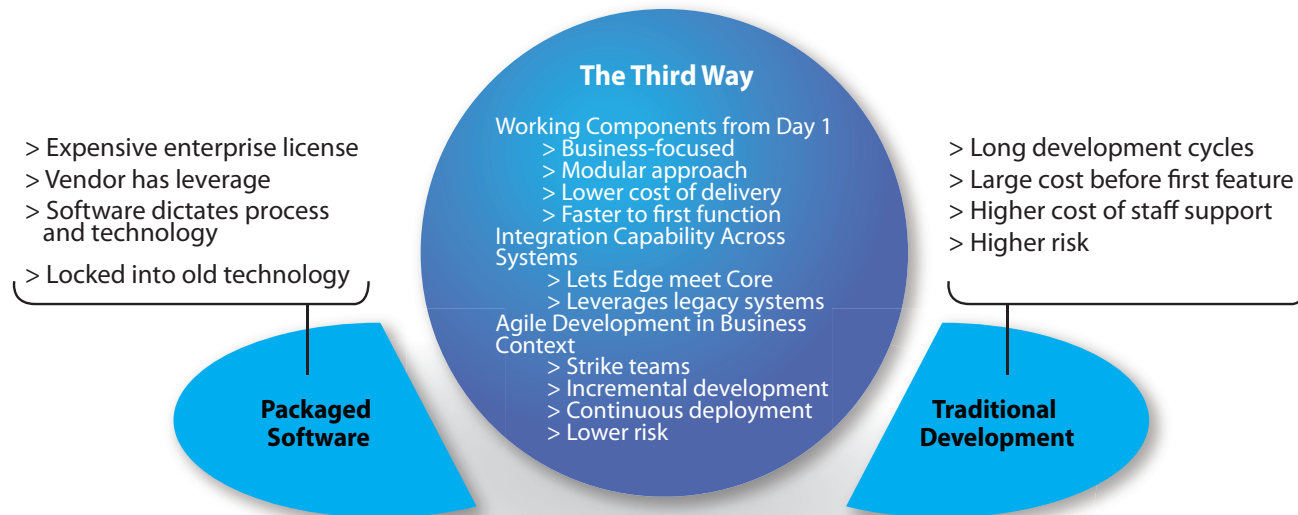
What does it mean to embrace these principles? It means that following them is the rule, not the exception, and that exceptions are difficult to get. Every IT initiative begins with asking how the deliverable and the method of delivery stack up against the principles. Most importantly, the *business* adopts these principles.

A systems and technology platform is a recurring business asset, not a one-time IT project. As a business migrates to its platform, the problems of "renegade systems" and "shadow IT" diminish – both because the platform is open to more local applications, and because IT can deliver new business capability so much faster.

## The Third Way

There have long been two basic approaches to putting business systems in place. Building custom applications from scratch was the norm until the rise of software packages and ERP systems with extensive functionality. Then most companies adopted a "buy, don't build" policy, with a strong preference to avoid customizing the software package. That preference notwithstanding, companies found that they had to modify the packages to fit local processes, and supplement them to provide differentiating functionality or simply a friendly interface. They then found themselves in a no-man's land – locked in to the vendor, its technology, and its software upgrades; and forced to do an ever-larger amount of custom redevelopment with each upgrade. It became a toss-up: Which is worse, dealing with vendor releases or continuing to maintain those ancient in-house systems and their COBOL code?

There is a third way. Build upon a common technological foundation including interface protocols and other standards. Treat everything – including existing applications – as modular components that should be exposed to the platform at large and reused wherever valuable. Use agile development methods to build and modify components. Automate applications configuration and interfaces to minimize the disruption of business and technology changes. You'll regularly start with packaged software, but keeping up with the vendor is simplified, and you have the option of simply making the software your own.



The third way entails a different applications development process. The first two steps of specification and programming combine. Instead of having a middleman called “business analyst” extract specifications, have business-conversant developers sit down with business people and prototype and exercise new business capabilities. Why talk about business processes and applications when you can work directly with them? We call the people “strike teams” and the process “side-by-side development.”

The last major step of applications development is deployment, including testing, installation and roll-out to user groups. With large legacy systems, production testing and installation of changes can each entail taking down the application, and user testing can send you back to the drawing board. However, with today’s technology, it’s relatively easy and inexpensive to create realistic test environments, and it’s possible to automate deployment to each of several environments: development,

testing, user testing, production. User testing is an extension of the strike team’s work. And as long as modifications are kept small and self-contained, and releases are frequent, the costs of introducing an error and making a fix are very low.

The result is unprecedented speed and quality. The specification step is accelerated by eliminating the middleman. Development is accelerated by agile methods drawing upon a platform of components. Deployment is accelerated through automation of testing and regularity of releases. And quality rises with a better brand of business involvement and more realistic testing.

This third way combines the implementation speed of software packages with the fit of custom development, adding inherent flexibility that is native to neither. Small teams accomplish more with less. The enterprise can think bigger and act faster.

**Grapevine** Stryve's methods for agile applications development, cumulative platform building, and unleashing the value of technology assets are called Grapevine. It's a unique combination of capabilities, therefore a bit tricky to explain:

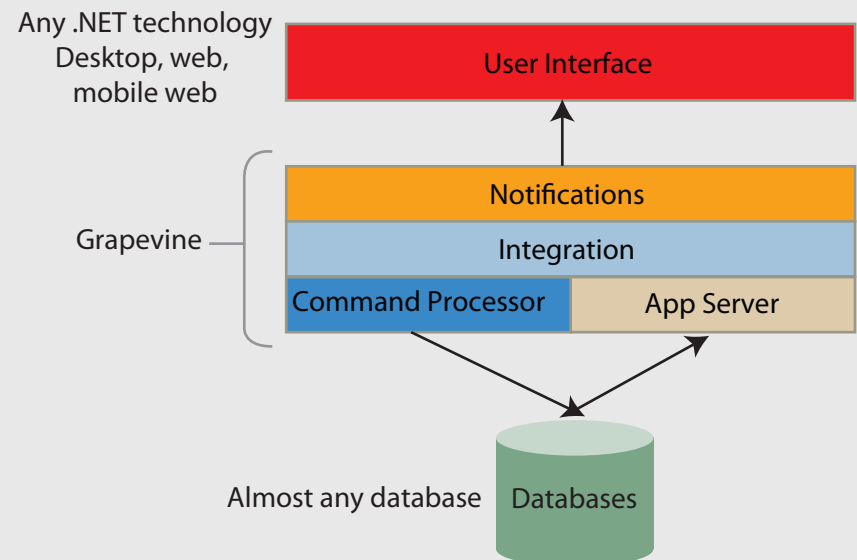
- It's not a proprietary technology, but rather uses powerful technology available to all.
- It's not a complete architecture, but it follows the architectural principles of a flexible platform.
- It's not a development methodology, but it really calls for a "third way" approach to applications development and deployment.
- It's not a productivity tool, but more an integrated set of technologies and techniques that yield unprecedented productivity.
- It's not a product, but rather a way of working with Stryve or on your own to tackle today's biggest information systems challenges.

For the technologists: Grapevine is a collection of .NET servers designed to run custom software. We use NServiceBus to connect the servers together. The code that performs business functionality needs to be written or repurposed by developers, but Grapevine helps them to structure the solution, provides basic infrastructure functionality, and acts as the host services that run their code in test or production environments. Grapevine provides guidance and automation for retrieving and saving data (including from other systems), updating user interfaces, and maintaining security.

In layman's terms, Grapevine provides the fabric for integration. The "plumbing" of interface management and other templates are predefined so developers can concentrate on business logic and data models and how they're used. Very importantly for platform integration, Grapevine documents and publishes what it's done.

Why "grapevine"? Think about running a vineyard. The terrain may be uneven, but the structure of rows and trellises is standardized, enabling consistent access for both workers and processes like irrigation. That corresponds to a common architecture enabling interfaces and automation. Individual vines correspond to applications. They follow their own patterns, but can be pruned or modified as they grow. The combination of common foundation and individual attention maximizes the productivity of each vine and the yield of the vineyard.

If the metaphor seems a stretch, just think of Grapevine as the flexible tendrils for extracting, modifying, revising, and repurposing data and code while applications go about their everyday duties. Grapevine is ultimately defined by its results – rapid implementation of new business capability and a way forward in overcoming the limitations of inflexible legacy systems.



## Better Methods at Work

The methods we've discussed prove valuable in meeting a wide variety of information systems challenges. Here are some examples.

**Rebuild fast.** An independent power producer acquired another producer, adding plants, expanding market coverage, and quadrupling the workload of power scheduling – a process at the heart of the business. With the application and tools in use, schedulers couldn't handle the larger workload in time to meet the daily market deadline. Rather than buying an application to address that one problem, the company invested in rapid (60 days) development of a new system that would also serve the changing needs of other scheduling, trading, and analysis activities. Because they could not anticipate all the potential uses, the developers created not a fixed application, but a set of functional capabilities with a common and intelligent underlying architecture. "Intelligent" because it keeps all data exposed to other applications, and it generates notifications to keep everyone up-to-date on changes to the system. The application's functionality has since changed and expanded with ease.

**Free the data.** An online marketing analytics company was founded by one of the top data modeling scientists in the U.S. Its core system was built using the best technologies of only a few years ago; however, with the explosion of demand for analytics and the rapid advance of computerized analytical methods, the system couldn't keep up. Rebuilding it would have constrained the company's growth. Instead, the company installed a Grapevine "adapter" into the existing system to continuously replicate its data in one of today's most advanced and high-capacity database managers. There the data scientists were able to develop new analytic functionality, including features they thought were still a year down the road. The old system

continued to run and support customers as usual. And the company had the start of a new architecture for future change. Many organizations have problems extracting data from legacy applications and databases, but few with the scale and urgency of this one.

**Franchise the applications.** A restaurateur with a successful regional chain wanted to expand by franchising. The company's competitive advantage lay in custom restaurant software for operational and financial management, a system that had been refined over a decade. To be deployed to franchisees as a SaaS offering, the functionality had to be rebuilt with newer technology. The application also had to be flexible enough to interface with dozens of popular point-of-sale systems, to enable mobile access for employees and managers alike, and to scale as franchises grew. The application was rebuilt in six months using third-way methods and a Grapevine architecture. It is easily provisioned to franchisees and, just as importantly, serves as a platform for continuous improvement, just as the original local system did.

**Surround the legacy.** Most companies will initially use these new technologies and methods to address the problems posed by inflexible legacy applications. There are several ways to "surround" an older system, letting it continue to run while putting its components to better use. Data can be extracted automatically and reformatted for decisions or analytics, dramatically reducing the manual effort spent generating reports. A new "skin" can be put on a legacy application to support simplified user interfaces or mobile access. New functionality can be developed in parallel then integrated as extensions of the application. Over time, that surrounding functionality can collectively replace old applications and their outdated (and often proprietary) technology.

Where should your company begin, or which application should it tackle next? Be sure to take two perspectives:

- **Problem.** What key application has fallen far behind what the business needs in terms of functionality? What application has become unmaintainable because its technology is old and the skills to maintain it are scarce? What application presents the most difficulty each time the software vendor releases an upgrade?
- **Opportunity.** What application will need to be more flexible in the future because it serves a very dynamic business need? What application has “buried assets” of great potential value to the platform at large? What application will unleash value by being “mobilized” to portable devices and remote users?

## Toward a New Normal

We’ve outlined and recommended new and better approaches to developing and managing information systems. Most companies we work with are applying some of them, but in bits and pieces. We argue that the time and technology are both right for a concerted effort, for changing the basic methods and attitudes of applications development. Even if your company has made a recent and large commitment to an ERP, this should still be the new normal:

- Existing applications, including older legacy systems, should not just be maintained, but be kept in play. Their data and business logic should be available for other uses, even as they are being gradually replaced.
- New application functionality can be developed quickly and before all the “requirements” are known. Any new application is changeable, scalable, accessible through a variety of interfaces, and integratable with other applications.

- Every new application and every change to an existing one migrates a little more of the computing platform to a superior and inherently flexible architecture.

What businesses need from their information systems and infrastructure continues to change fast. Today’s premium is on speed and innovation. Fortunately, technologies and techniques have also advanced fast – to the point where we can implement much more flexible and all-purpose information systems platforms. Businesses can have consistency, scale, and operational performance where they need it, and they can put many of the same technology assets to use to experiment and innovate. We can’t know what changes we’ll face tomorrow, but we can make our most important information systems much more change-able. We can answer those six questions in the affirmative.

## ABOUT THE AUTHORS



**Jim Champy** | Vice Chairman | Stryve

Jim Champy is a leading authority on the management issues surrounding business reengineering, organizational change, and corporate renewal. He is the former Chairman of Consulting for Perot Systems and was a founder of Index Systems. Jim consults extensively with senior-level executives of multinational companies seeking to improve business performance.

Jim is co-author of *Reengineering the Corporation*, which was on the *New York Times* best-seller list for more than a year, was translated into 17 languages, and sold more than two million copies. His other books include *Reengineering Management*, *Reengineering Health Care*, and the recent series *Outsmart!*, *Inspire!* and *Deliver!*



**Colby Thames** | Managing Partner | Stryve

Colby Thames has delivered insight, research, and technology solutions to Fortune 500 enterprises since 1990. As a founder of Stryve, he is responsible for its overall strategy and corporate operations. In addition, Colby provides executive guidance on critical issues to many of Stryve's clients.

His research work centers on corporate challenges at the intersection of information technology, process, and human capital. His most recent research has explored social business process reengineering, next generation of shared services, and corporate collaboration.



**Jay Grieves** | Chief Technology Officer | Stryve

Jay Grieves is an innovative technology consultant and software developer specializing in development practices and large-scale systems. His client list includes some of the largest and most technologically sophisticated companies in the energy and financial sectors. A technology professional for more than 15 years, he has worked for major organizations including AARP and Microsoft Research, and he served as CTO with DuClaw Brewing Company. In 2004 he formed Brightfly Development (now part of Symantec) to build pioneering compliance management software.



**Robert Morison** | Research Director | Stryve

Bob Morison is a business researcher, writer, discussion leader and management consultant. He has been leading breakthrough research at the intersection of business, technology and human asset management for more than 20 years, and has developed management guides on topics ranging from workforce management and business innovation to business process reengineering, collaborative business models, and business analytics.

He is co-author of *Analytics At Work: Smarter Decisions, Better Results*; *Workforce Crisis: How to Beat the Coming Shortage of Skills And Talent*; and three *Harvard Business Review* articles, one of which received a McKinsey Award as best article of the year.



From conception to execution of technology-enabled business change, Stryve provides a full range of adaptable services, collaborates with clients on program and project management, and drives both high ambition and superior results.

**For further information, please contact  
Ron Christman, Managing Partner, at 281-536-6490 or [Ron@istryve.com](mailto:Ron@istryve.com).**

