

CIO STRATEGIES FOR SUCCESS

Choosing the right strategies, resources, and software





Table of Contents

Executive summary	3
Threats to CIO Job Security	3
Poor Performance	3
Major Application Failure	4
Compliance Failure	4
Missed Deadlines and Exceeded Budgets	4
The Disconnect Between Business Managers and IT	4
Strategies Ensuring CIO Success	5
The Right Management Approach	5
Leadership Skills	6
Management and Communication Skills	6
Alignment with Business Executives	6
The Right Software	7
Vendor Checklist for an Enterprise-Worthy BPM Solution	7
Innovative Development Methods	9
Reducing Project Development Times and Risks	10
Code-Free Development Contributes to CIO Success	13
Sample Applications Built With Agiloft	14
Agiloft Technology and Platform	15
Conclusion: The CIO as Hero	15
About Agiloft, Inc.	15



EXECUTIVE SUMMARY

CIOs may have the toughest job in IT. After all, they're responsible for customizing, integrating, deploying, and supporting hundreds of applications. Simultaneously, they must meet increasingly tough standards for auditability, up time, and security, while reducing costs and simplifying life for their business users.

So it's not surprising that their initiatives are often perceived as falling short. Neither should it come as a shock that their average job tenure is less than five years. Yet many of the challenges faced by CIOs can be anticipated and neutralized or eliminated entirely through a combination of business strategies and appropriate technologies and vendor relationships.

This paper describes the threats to CIO job security and suggests strategies for addressing them. It argues that a major cause of perceived CIO failures is that IT professionals and business managers have such different expectations for how long things should take, that IT may be perceived as non-responsive even when it is doing a superb job with the available technology.

We examine some management approaches of successful CIOs, who commonly exhibit strong leadership and management skills, as well as the ability to align themselves with their business leaders. Despite differing time-lines and expectations between IT and business departments, these CIOs deliver the software solutions their business leaders need, when they need them.

We then address the critical issue of choosing the right software solutions to meet the needs of business users in a timely manner, without sacrificing scalability, reliability, and functionality. After providing a checklist of essential attributes for an enterprise-worthy solution, we compare and contrast different development approaches and their impact on development time and responsiveness.

We examine the time savings provided by the innovative methodology of code-free development in more detail. Project time lines using this approach are contrasted with more traditional approaches. An analysis of project stages illustrates why the new methodology reduces deployment times by a factor of two or more.

The use of resources, in particular, in-sourcing versus outsourcing, is also considered as a means of reducing deployment times. We lay out some pros and cons of using internal IT staff, external contractors, and outsourcing.

A description of the benefits of code-free development and its potential contribution to CIO success is followed by examples of deployments of Agiloft at Fortune 500 and smaller companies.

THREATS TO CIO JOB SECURITY

Let's begin with the "worst case" scenario: when an IT department does not satisfy its company's business needs, the CIO is typically let go. By examining the most common reasons for dismissal, we can draw some conclusions about how they can be avoided.

Poor Performance

As summarized in a CIO.com¹ article, the primary reason why CIOs are fired is "poor performance." Surprisingly, this charge can apply even when the IT department has successfully rolled out major new applications on time and on budget². Such a deployment may be a success from an IT perspective, but if the application does not provide ROI to business users, it will still be deemed a failure.

In many cases, CIOs may be using different metrics for success than business users, and so are blind-sided by such "failures." As Laurie Orlov, a principal

¹ Kim Nash, "One in Four CIOs Fired for Performance," *CIO.com*, March 11, 2009, www.cio.com/article/484008/One_in_Four_CIOs_Fired_for_Performance

² Renee Oricchio, "Leading Reasons Why CIOs Get Fired," *CIO Strategy Center*, www.ciostrategycenter.com/wjz/Board/peers/leading_reasons_why_cios_get_fired/index.html

analyst from Forrester Research notes, “CIOs need the ability to run their IT departments in a business-like way. Too many can’t talk the same metrics as their colleagues on the business side.”

The key to avoiding this trap is to get close to the business users and actively monitor progress. For example, online surveys can provide insight into their perspectives and warn of pending issues and frustrations. Subsequent surveys enable the CIO to measure progress and may help top management recognize improvements.

Major Application Failure

When a major application fails, the IT department takes the blame, even if the cause is some unanticipated circumstance, such as an exceptionally high load. Systems must therefore be scalable to several multiples of the anticipated loads and fully redundant. Additional recommendations are detailed in the “Vendor Checklist” section of this document.

Compliance Failure

Since the advent of Sarbanes-Oxley, failures in this area can mean not just dismissal, but lawsuits and even jail time for top executives. It is not enough to demonstrate that appropriate procedures exist; it must be possible to show an auditor how they were followed in any particular instance.

Tracking compliance manually is a costly and unreliable endeavor. A more solid approach is to build compliance into relevant business processes and the software tools used to automate and manage them. As a practical matter, this requires software that incorporates compliance as core, not bolted-on, functionality. The goal should be to build systems that are fully audit-capable because all relevant user actions are tracked and recorded automatically. Depending upon users to enter audit information is costly and error prone.

Missed Deadlines and Exceeded Budgets

There are countless examples of such failures, ranging from Kmart scrapping a software distribution system³, to cost overruns in the state of Virginia’s outsourcing deal with Northrop Grumman⁴. Perhaps most alarming is The Standish Group’s assertions in their “Chaos Report” that fewer than 35% of projects are considered a success, and roughly 25% are cancelled.

The key to eliminating such problems is to remove unknowns as far upstream as possible. While it may not be possible to nail down every detail of a large deployment in advance, it is possible to structure spec development and implementation milestones so that they are uncovered and resolved before significant resources are committed. The “Innovative Development Methodologies” section of this document describes approaches by which development time frames can be reduced and unknowns eliminated for many projects.

Where third parties are involved, it is also desirable to structure the relationship so that the contractor bears full responsibility for cost containment and delivery time-frames. The “Vendor Checklist for a Responsive BPM Solution” section of this document describes the attributes of a software system and vendor relationship that make this possible.

The Disconnect Between Business Managers and IT

In many cases, a major source of frustration for business managers is that they and IT departments have completely different time scales. When business managers request a change to an application that

³ Renee Oricchio, “Leading Reasons Why CIOs Get Fired,” *CIO Strategy Center*, www.ciostrategycenter.com/wjz/Board/peers/leading_reasons_why_cios_get_fired/index.html

⁴ Paul McDougall, “Virginia Probes Outsourcing Deal, CIO Fired,” *Information Week*, June 2009, www.informationweek.com/news/government/state-local/showArticle.jhtml?articleID=218100342

supports their business processes, such as a change in the workflow, they believe it should take no more than a few days. On the other hand, IT departments typically regard a turnaround time of two months as fast. When a new software package is being deployed, business managers would like to see it up and running in a matter of weeks. However, in reality, Business Process Management (BPM) projects often have timelines exceeding nine months, even in standard areas, such as customer support or change management.

This is so fundamental a difference that CIOs are often shocked to find out how much the IT department's deployment schedules are resented. As a result, they're blind-sided by charges of poor performance.

This really came home during a presentation on "Software as a Service" (SaaS) BPM technologies to a group of MBAs. During the talk, the presenter said he suspected "one of the reasons they were choosing SaaS vendors [as opposed to their in-house IT department] was that the people who caused them the greatest frustration were not their external competitors or their customers, but rather..."

Before he even had a chance to finish the sentence, the entire class yelled "**The IT Department!**" Well, the entire class apart from two startled IT guys. In his department's defense, one of these guys remarked that his users must be happy because they rolled out a new iteration of a certain system every three months. The MBAs crucified him: "Three months! Don't you understand anything? When I need a change, I need it the next week. Maybe I can live with two weeks, but three months?! And you're proud of that?"

Within companies, business managers are the customers – and a three-month turnaround time for changes is not "quick" in their terms, even though it may be positively heroic with

certain systems. When IT fails to meet business requirements, the CIO takes the heat. This happens so often that some joke that the acronym "CIO" has come to mean "Career Is Over."

Chris Patrick, who runs the global CIO practice group in Dallas for executive recruiter Egon Zehnder says, "I'm seeing a lack of patience for CIOs who can't deliver results quickly. They hired you to have impact. You can't go in and lay out the five-year program that starts in year four."

Though those of us in the software trenches realize how unrealistic business managers can be, we must concede that they have a point. Quick IT project deployment or modification in a rapidly changing business environment is imperative.

When Charles Darwin uttered the words, "**It's not the strongest, nor the most intelligent that survive, but the most adaptable,**" he likely had no idea how much they applied to corporations. Yet as we've seen, the entire economy can go from boom to bust in less than six months, often necessitating a 180-degree flip of business priorities. Unfortunately, many corporations are stuck with inflexible, difficult-to-change technologies that hamper adaptability and carry high maintenance costs. With such a long lag time for deployment or modifications, software is usually out of sync with current needs.

STRATEGIES ENSURING CIO SUCCESS

Now, let's look at some solutions to these challenges. Two key strategies involve using effective management approaches to stay ahead of the curve while making the right software decisions to ensure the company's success.

The Right Management Approach

A successful CIO must move beyond putting out fires to become a key contributor to the business. This requires specific people and communication skills as well as an understanding of how to gain the respect and cooperation of the other high level business managers. Let's examine these skills more closely.



Leadership Skills

"A CIO has to demonstrate the right level of sophistication for the job. A lack of leadership will be the first thing to expose a CIO's incompetence," says John Stevenson, a former president of the Society for Information Management, a professional organization for IT leaders.⁵

As Gartner points out, "Leadership and management are different, but complementary. Management is about execution. Leadership is about change." CIOs must "influence and lead their business colleagues by influencing their view of IT."⁶

This task is a lot easier if the changes being proposed are clearly desirable from a business perspective. For example, if the CIO can propose changes that shorten application deployment times, reduce complexity, and improve IT responsiveness, buy-in will be immediate and enthusiastic. Demonstrable success in achieving such goals will give the CIO the credibility and political capital necessary to institute other changes whose benefits may not be so immediately apparent. The "Reducing Project Development Times and Risks" section of this paper describes a methodology for achieving such goals.

Management and Communication Skills

CIOs must not only manage IT staff, contractors, and vendors, but also ensure that they work effectively with business users. This requires people skills, including effective and pro-active

communication.⁷

Regular communication is essential to CIO success but can be time-consuming and prone to falter when crises and time-crunches occur. While some communications are necessarily verbal, many written communications can be automated through the development of clear business communication processes and the help of an effective process automation tool. Putting this kind of system in place can be a lifesaver, since ensuring that the right people are properly informed is essential to setting accurate expectations and enabling business requirements to be fully addressed in technology projects.

Developing automated communication processes saves time and eliminates human errors. For example, a centralized project/change management system can automatically email reports on project statuses to the appropriate stakeholders, keeping business managers informed about the technology projects that affect them. Automated alerts can also provide pro-active notifications of pending tasks and, where necessary, escalate issues and re-assign tasks to ensure that they are completed before they impact delivery time frames.

Alignment with Business Executives

"It's essential today that a CIO knows how to fit like a glove with the other C-level executives and other influential leaders within the organization," says John Stevenson.⁸

Simply making sure that the servers stay up is not enough. The IT department must provide a tangible competitive edge from the perspective of the business users. This requires the ability to respond to changing market conditions, implement management

⁵ Renee Oricchio, "Leading Reasons Why CIOs Get Fired," *CIO Strategy Center*, www.ciostrategycenter.com/wjz/Board/peers/leading_reasons_why_cios_get_fired/index.html

⁶ Marianne Broadbent and Ellen Kitzi, "The New CIO Leader: Setting an Agenda and Delivering Results," Gartner Group, www.gartner.com/5_about/news/gartner_press/NewCIO2.jsp

⁷ Carter McNamara, "Basics in Internal Organization Communications," *Field Guide to Leadership and Supervision*, http://managementhelp.org/mrktng/org_cmm.htm

⁸ Renee Oricchio, "Leading Reasons Why CIOs Get Fired," *CIO Strategy Center*, www.ciostrategycenter.com/wjz/Board/peers/leading_reasons_why_cios_get_fired/index.html

initiatives fast enough to be seen as responsive, and exploit opportunities while they are still available. In brief, it means agility.

The Right Software

This kind of agility is probably the single most important characteristic that distinguishes embattled CIOs from their more respected and successful colleagues. How can it be achieved? By ensuring that the software used to manage the major business processes of the company is agile — that is, easy to modify as needs change, and quick to deploy.

Business Process Management (BPM) software solutions manage and automate a wide range of business processes - everything from internal and external customer support to sales, marketing, change control, RMAs, assets, employee expense reimbursement, hiring, HR requests, time keeping, billing, contracts, projects, and so on. Process automation improves efficiency, reduces manual error, ensures compliance, and increases reliability, but processes change frequently as the business environment changes. Thus the need for agility.

To be perceived as responsive to business needs, a BPM solution should enable significant projects to be built and deployed in weeks and modified in hours or days – a timeline to please even the most demanding CEOs.

Vendor Checklist for an Enterprise-Worthy BPM Solution

However, fast is worthless if it also means shoddy, expensive, or unproven. So we must begin our discussion about choosing the right software with a list of technical and business requirements that a system for enterprise use should satisfy.

Auditability

The system must be auditable in multiple senses to ensure compliance with Sarbanes-Oxley and other regulations. It must make it easy to show an auditor what a defined business process is,

how the system enforces the process, and how the process has been followed in any particular instance. Further, the solution should make it possible to capture and collate data, such as who logged in, what IP address they came from, what records they viewed, edited, etc.

Integration

The solution should include prebuilt integration with standard technologies, such as LDAP/Active Directory and MS Exchange. It should also support a robust set of APIs and scripting options, including Web Services.

Adaptability

Once the system has proven itself in the initial deployment, it should be easily extensible to other business areas. So, the data models, business rules, workflows, access permissions, and data input forms must be fully and rapidly customizable.

Scalability

The solution must scale to support thousands of current users, the update of hundreds of thousands of records per hour, and databases containing tens of millions of records, without requiring non-commodity hardware.

Web-Based to Reduce Maintenance

The product should be 100% Web-based so that no installation or upgrading of client software is required. It must support the company's choice of browser.

Security

The system must support a fine-grained security model for precise access control. The software platform and hosting infrastructure (if SaaS-based) should be subject to regular security audits from an independent firm, and the vendor should make the results available.

Uptime

For SaaS-based products, vendors should provide up-time guarantees reflecting their confidence in the availability of the service. Some vendors just offer a

pro-rata refund, while others return the entire cost of that month's service if the target up time is not met. If the product is installed in-house, it should support high availability options so that service can continue even in the event of a motherboard failure.

Reporting

The system must support dashboards, charts, and reports providing quick insight into business processes. Since passive access to information is not always enough, it should also support the creation of business rules providing active notification of any problems.

Standards Compliance

The system should support standards such as HIPPA, ADA, ITIL, and CFR 21 Part 11.

Platform Choice

The vendor should offer a SaaS option so that customers don't need to provision a server to get going. Once the solution has proven itself, it should be movable to their choice of in-house Linux or Windows server to allow full integration with sensitive back-end systems without impacting the firewall.

Backups

System backups should be fully automated and include everything necessary to move the entire deployment to another server or restore in case of disaster.

Upgrades

Upgrades should require little effort and must allow migration from any revision to any later revision without affecting customizations.

Cost

The cost to get started must be reasonable and the product should provide a rapid ROI, ideally within the first few months of use. Getting a reasonably complex production system up and

running should take a matter of weeks, and extending it to cover new processes must be equally rapid, without impacting existing production use. The cost structure should be simple and inclusive, without hidden extras or fees when the system is extended.

Vendor Independence

IT staff should be able to extend and maintain the system themselves after training. Customers shouldn't be tied to long-term dependence on \$200-per-hour consultants. Ideally, the training time should be short. Systems designed to be maintained by the users may require a week of training to reach proficiency, whereas those designed without this criterion in mind may require over a month of training and carry increased effort and risks when making changes.

Company Stability

The vendor should have a ten-year or more history of providing enterprise solutions. For CIOs of large companies, the vendor's track record with other Fortune 500 companies is most relevant. For start-ups, experience with small companies is of greater interest. The vendor should be financially sound and profitable.

Low Risk

The vendor should be able to describe exactly how the software addresses current business need(s) and demonstrate it running this exact process prior to purchase. It should also be willing to commit to a fixed-price implementation for the entire project based on an agreed upon specification.

Different vendors may offer different forms of refund if a project fails, ranging from a credit towards additional software purchases, to a full cash refund of all software costs and consulting services. The strength of the warranty indicates the vendor's confidence in their software and implementation services.



Conclusion

Different CIOs may prioritize their requirements differently from this list, but it should serve as a useful starting point. The extent to which these requirements are supported varies widely among vendors. The manner in which all of these

requirements are met by Agiloft is detailed on the Vendor Checklist page at our [website](#).

Innovative Development Methods

Now that we've established some essential quality requirements for a software solution, let's

Methodology	Pros and Cons
Integrated Development Environments and Scripting Customizable Prebuilt Applications	<p>Pro: Environments such as .NET and J2EE and scripting languages attempt to reduce development times and provide total flexibility.</p> <p>Con: In practice, it still takes so long to develop significant applications it is only affordable by software companies that spread them out over multiple customers.</p> <p>Pro: Software companies provide pre-built applications that can be extended through custom coding so that customers can tune the application to their specific needs.</p> <p>Cons: This solution is only applicable if the pre-built application is reasonably close to the customer's requirements. It may be easy to extend an application through custom code, but hard to change the pre-built/hard-coded behaviors.</p> <p>Many customers underestimate the cost of maintaining the custom code throughout the life of the project. Yet, development only accounts for 20% of the total cost of a typical software project. Eighty percent of costs come from bug fixes and "minor enhancements" during production use.</p>
Cloud Hosting Environments	<p>Pro: Cloud offerings, such as AppExchange, EC2, and App Engine, provide a hosting environment for applications. By leveraging this infrastructure investment, companies are able to offer a highly reliable service with low upfront costs.</p> <p>Con: This solves the physical infrastructure problem, but customers still have to build or buy/customize/integrate applications and pay ongoing monthly charges for an infrastructure that lives outside their firewall.</p>
Code-Free Development Environment	<p>Pros: Code-free development environments allow applications to be developed and customized using just the browser. As detailed below, this typically takes less than half as long as it takes to customize traditional prebuilt applications, and applications built on the platform are fully integrated. When custom coding or integration with 3rd party software is desired, standard languages such as Java, Perl, or Web Services may be used.</p> <p>Sample applications developed without coding include: CRM, ITIL, Helpdesk, Sarbanes Oxley, Change Control, Asset Management, COCOM, Procurement, Provisioning, RMA Management, Contract Management, Equipment Reliability and Performance Tracking, HR and Time Management, and Event Management.</p> <p>Cons: The range of applications that can be developed without the addition of custom code is limited by the core capabilities of the system. Although this has not been an issue to date, there must be applications that would require custom coding or core enhancements.</p> <p>Although the back-end data models and workflows are fully customizable, the GUI is automatically generated so the resulting applications tend to look alike. This might be seen as a plus by business users who dislike having to learn or use multiple interfaces, but it restricts the creative freedom of developers.</p>



examine and evaluate the various means by which software companies attempt to reduce project development times, since software responsiveness and agility are critical to success.

The availability of a code-free development environment that supports sophisticated enterprise applications is a significant development and is examined in more detail in the next section.

Stage	Activity	IT Effort
Functional Requirements Specification (FRS)	Define the software requirements from the user perspective.	1
System Requirements Specification (SRS)	Define the manner in which the user requirements will be implemented within the chosen framework so as to satisfy requirements for scalability, security, auditability, etc.	1-3
Implementation	System implementation may consist of raw coding in an environment such as .NET or SQL, or customization of an existing package, such as a CRM system, through callbacks. For projects involving complex, traditional software, these numbers may go much higher.	4-20
Unit Testing	Test each part of the system, typically carried out by Engineering to confirm that it meets the SRS.	2-6
Initial Production Use	Test the entire system, typically carried out by a small group of users, in a "trial mode." Note: Ideally, only small changes/fixes are required, but it is not uncommon for users to find that the SRS did not meet the actual business requirements and the system is simply unusable in production, resulting in a new round of spec development and implementation. The estimates in the right column therefore represent a best-case scenario.	2-6
Production Support	Support production use of the system with bug fixes, upgrades, and modifications to reflect experience and changing requirements. Notes: As detailed by The Standish Group's "Chaos Report," approximately 25% of projects are cancelled before they reach this phase.	10-20
Decommissioning	This is the time taken to transfer the business processes from an obsolete or difficult-to-maintain system to a replacement system. Note: The original developers have typically left the company by this time, which can make the transfer more challenging, especially if the system documentation has not been kept up to date. Unless one knows exactly what the current product is doing and how it is doing it, the application is hard to replace. This is one reason why so many COBOL applications are still running on mainframe computers.	2-10
Totals	Minimum: 22 and Maximum: 66 Man-Units	



Reducing Project Development Times and Risks

As we have seen, one of the most critical areas of contention between CIOs and their business users is the long timelines necessary to roll out and make user-requested changes to critical business applications. Becoming more agile and responsive in this area can probably do more to enhance a CIO’s success than any other change.

In this section, we compare traditional IT project development and man hours with a code-free development platform, using our decade of experience implementing Agiloft as the model for code-free development time frames.

Traditional IT Project Stages

IT projects typically go through the phases outlined in the table below, with the nomenclature varying from company to company. The time taken depends upon the project, but the ratio of time spent in each stage

Stage	Activity	IT Effort
Functional Requirements Specification (FRS)	Define the software requirements from the user perspective.	1
SRS	Scalability, security, auditability, etc. are all built into the system and there is rarely any code to write, so the need for a traditional SRS is greatly reduced. However, it is still necessary to define the data model that will provide the most intuitive user experience.	1
Implementation	Implementation consists of configuring tables, business rules, workflows, etc. using a browser. With no code to write, this takes about as long as the FRS and SRS took to design. The need for unit testing is eliminated because the core code has already undergone tens of millions of hours of production use.	2
Initial Production Use	Test the entire system, typically carried out by a small group of trial users. Note: Since it is possible to make changes so rapidly within the code-free environment, it might be expected that this phase would take much less time than using a traditional methodology. However an effective trial still requires significant man-hours, and the ability to make changes on the fly encourages users to fine-tune and extend the system more than they would in a traditional environment.	2-6
Production Support	Support production use of the system with modifications to reflect experience and changing requirements. The time required for changes is much less than with traditional methods because there is no code involved. Note: Unlike traditional IT projects, no Agiloft- based development has failed to reach production use.	2-6



Decommissioning	Transfer the business processes to a replacement system. Note: This figure is estimated because, to date, customers have extended, not de-commissioned, Agiloft deployments. However, the most serious challenge for project transfers is eliminated because the system is self-documenting. Every data structure, business rule, and workflow is exposed through the administrative interface using the browser.	1
Totals	Minimum: 9 and Maximum: 17 Man-Units	

is less variable. So, if one week is spent detailing the user requirements, then between four and 10 weeks are typically spent implementing them. The IT Effort column shows the typical proportion of work (not elapsed time) time spent in each phase, whether this is in man-weeks, months, or some other unit.

Code-Free Project Stages

We have found that the timeline for code-free development is quite different:

In summary, if we look at the average development time, taking the middle of each range and assuming that the FRS phase took one man-week, we have the following comparison between the two methodologies:

The actual time taken in each stage depends upon the size and complexity of the project, but the overall ratios will remain about the same. In addition to reducing the time required to reach initial production use by a factor of three, the chance of failure is effectively eliminated.

Another Approach to Reducing Time: In-sourcing Versus Outsourcing

When it comes to reducing timescales and costs, there are multiple approaches. Where staff resources are limited, the use of consultants or outsourcing may bring faster results. For instance, they may enable a

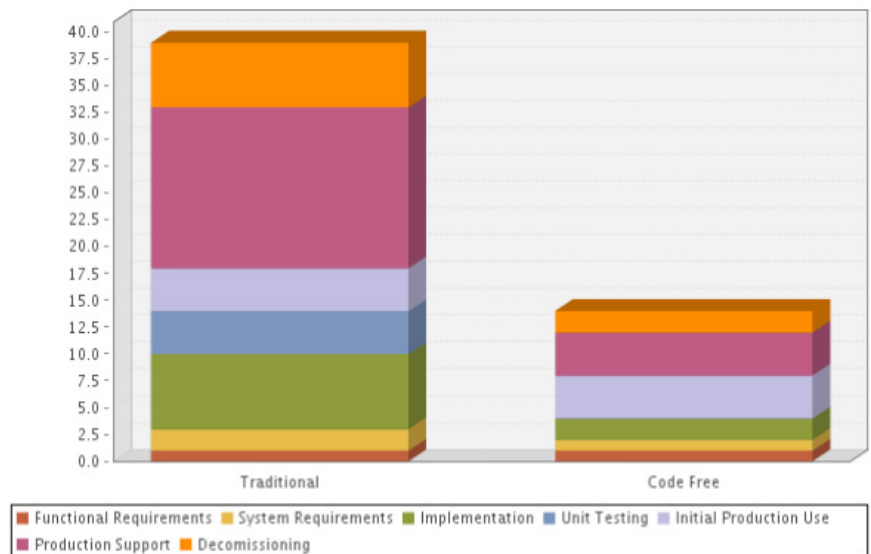
company to bring new business processes online more quickly than they could do otherwise.

A 2009 Gartner CIO survey⁹ reported that CIOs' top three priorities were implementing business process improvement measures, reducing enterprise costs, and improving workforce effectiveness. CIOs typically apply a judicious mix of resources to achieve these goals. Below, we summarize some of the pros and cons of using internal IT resources, consultants, and outsourcing companies.

Resource Types

Since each type of resource comes with significant pros and cons, the decision for how to mix them must depend on the priorities of an organization and the project at hand. For instance, outsourcing a short-term project can minimize costs, but may not

Man-Weeks Effort During Project Stages





Resource	Pros	Cons
Internal IT Staff	<p>Tenure: IT tenures are reasonably long. As a result, the people responsible for development are available to support modifications based upon initial user experience and changing requirements during the first few years of production use.</p> <p>Reliability: IT staffers are known quantities, with proven expertise, training certifications, and productivity.</p>	<p>Tenure: The longevity of enterprise systems typically exceeds that of IT staff, so extensive documentation must be supported for any custom code.</p> <p>Flexibility: It takes a significant amount of time to hire and train internal staff.</p>
Specialized Contractors	<p>Specialized Expertise: Contractors can provide expert and time-saving advice in particular areas, particularly when they have domain expertise that IT staff may lack.</p> <p>Flexibility: Contractors may be hired and fired at will.</p>	<p>Cost: The costs of highly qualified contractors can be double or more that of regular staff.</p> <p>Tenure: Responsibility for maintenance of the contractor’s work must be transferred to regular employees, and this knowledge transfer is frequently imperfect.</p>
Outsourcing	<p>Cost: Outsourcing companies can still provide resources at a much lower rate than local IT staff. The difference in pay between the U.S. and offshore suppliers is diminishing, but is expected to be significant until 2015 or later.</p>	<p>Expertise Loss: Core expertise may be lost, leaving the company dependent upon a supplier with its own set of interests.</p> <p>See this Gartner report for advice about avoiding specific outsourcing mistakes.</p> <p>Communications: Managers are challenged by different time zones, cultures, and languages, as well as intermittent connections.</p> <p>Tenure: The outsourcing company may re-assign staff to a better paying client.¹⁰</p>

be appropriate for a long-term, mission-critical project that requires ongoing maintenance.

If outsourcing is being used primarily to provide a solution or workaround to the cost and long development times required for process improvement and automation, it may become unnecessary once an adaptable and quickly implemented BPM solution is available.

Code-Free Development Contributes to CIO Success

A code-free solution like Agiloft helps avoid the most common causes of a CIO’s perceived failure. Let’s re-examine the main reasons why CIOs are fired, as described by the aforementioned *CIO Strategy* article¹¹:

- **The project never gets finished or goes too far over budget.** Removing the need for custom coding reduces the time required to develop a project by a factor of two or more. To meet your budget, you may do the implementation yourself or let us do it on a fixed price basis, based on a mutually agreed specification.
- **Major application failure.** Using Agiloft is very different from using unproven custom code — it is a code set that has been tested for scalability, audited for security, and proven in hundreds of enterprises worldwide. This is why we can guarantee success; none of our implementations has ever failed.
- **Non-compliance or a high-risk issue compromises the organization.** Compliance support is built into every Agiloft application. It not only shows an auditor what a defined business process is and how the system enforces it, but how the process has been followed in any particular instance. It also captures and collates data, such as who logged in, what IP address they came from, what records they viewed, edited, etc.

Furthermore:

- User adoption is a lot easier with a system that can be rapidly adjusted based on user feedback.
- There are no code-compatibility issues with upgrades because there is no custom code.
- Business managers no longer need to agree with one another on everything six months in advance. After all, the system can be changed using just a browser in a few hours. They are also no longer dependent on the “common sense” of programmers to deliver the system they need.
- The system is self-documenting because everything is exposed through the admin’s browser and graphical workflow editor.

- Data integrity is automatically maintained by the system, not by custom code.
- Code maintenance accounts for 80% of the cost of the average software project. With no code to write, there is no code to maintain. As a result, added costs, hassles, and unpredictable delays are eliminated.
- Multiple applications can be consolidated on a single platform.

Sample Applications Built With Agiloft

With Agiloft, applications may be developed from scratch or by customizing (and extending) an existing template or application. Typical deployments of the latter usually involve one week of consulting services or one week of training, if administrators wish to do the work themselves. Here are some examples showing how quickly applications can be built from the ground up, *without writing a line of code*:

- In just three man-months, we developed Agiloft CRM, a robust system being used by hundreds of companies worldwide, which has displaced major CRM products, such as MS CRM. See www.agiloft.com/hottopic-case-study.pdf to learn more.
- In just two months, we developed a Sarbanes Oxley application for Chevron. Learn more by visiting www.agiloft.com/chevron-case-study.pdf.
- In just six weeks, we developed a system that completely replaced a highly customized Vantive implementation. This included importing all the data, reproducing the data relationships, and re-creating the business rules. Since deployment, one staffer has single-handedly extended the system to cover problem, event, asset, and contract management *without any coding or consulting services*. See www.agiloft.com/carestream-case-study.pdf for details.
- In just two months, we developed a robust ITIL system. See www.agiloft.com/itil.htm to learn more.

A wide range of prebuilt Agiloft applications are available as starting points for custom applications and can typically be fully configured and deployed in a matter of weeks.

¹¹ Renee Oricchio, “Leading Reasons Why CIOs Get Fired,” *CIO Strategy Center*, www.ciostrategycenter.com/wjz/Board/peers/leading_reasons_why_cios_get_fired/index.html

The last section showed how code-free technology reduced the time necessary to customize and deploy an existing application by an approximate factor of 3. As the above examples illustrate, the time to build an entirely new application is reduced by a far greater factor. For example, Microsoft CRM represents several man-centuries of effort, yet was displaced by an Agiloft-based application that was developed in 3 months. For enterprise applications that are highly specialized, the actual time savings will therefore be far greater than 3 to 1.

To watch a demonstration of building an application from scratch, visit the following page at our Web site: www.agiloft.com/flash/Building-a-custom-application.html. This presentation literally walks you through building a brand new application for managing employee travel expenses in just 25 minutes. It's a bit boring, since it shows every mouse-click and keystroke involved. But from a technology perspective, it's quite interesting. After all, the resulting application includes full Web services support for both .NET and Java, support for thousands of concurrent users, the generation of over 200,000 records per hour on a \$5,000 server, and more. In fact, it includes everything described in the "Vendor Checklist" section above. In brief, it is a true enterprise application.

Agiloft Technology and Platform

So, how does our technology enable this rapid, code-free development of Web-based enterprise-class applications? Agiloft leverages open-source technology stacks, the dynamic capabilities of J2EE, and the availability of commodity hardware with 12-plus gigabytes of RAM. Of course, if you are the rare customer who needs some custom scripting, it also supports standards such as JavaBeans, Perl, and Web Services.

You've never heard of us? Well, we've been busy. We've been delivering enterprise software for 18 years and began development of the adaptive Agiloft platform over twelve years ago. It took us several man-centuries, five elapsed years, and over two million lines of J2EE code before we began to put it through its paces building solutions. The result makes sophisticated operations look easy, and it changes the entire equation.

If you are working to improve and automate your processes, addressing issues of compliance and control, or implementing metric based management, we can probably help. Call us at 1-888-727-2209 and we will automate your most cumbersome process for a personalized demo.

CONCLUSION: THE CIO AS HERO

An agile BPM solution can provide the ability to profit from new opportunities as soon as they occur or to follow new business directives as they are formulated. With Agiloft's code-free development environment and pre-built applications, CIOs can empower their companies to adapt to changing needs in hours or days, rather than months or years. The result: business managers who award their CIOs an A for adaptability. Effective BPM means success for the CIO and the rest of the organization.

ABOUT AGILOFT, INC.

Over 3 million users at organizations ranging from small enterprises to U.S Government agencies and Fortune 100 companies depend on Agiloft's top rated product suites for [Contract Management](#), [Service Desk](#), [Custom Workflow](#), and more. Agiloft specializes in automating processes that are too complex for competing vendors. Our best practice templates and agile technology ensure rapid deployment and a fully extensible system. For more information, visit <https://www.agiloft.com>.

