

Forrester Consulting

HELPING BUSINESS THRIVE ON TECHNOLOGY CHANGE

The Challenges Of Software Change Management In Today's Siloed IT Organizations

A Commissioned Study Conducted By Forrester Consulting On Behalf Of Serena Software

November 2006

FORRESTER®

FORRESTER®

Headquarters

Forrester Research, Inc., 400 Technology Square, Cambridge, MA 02139 USA
Tel: +1 617/613-6000 • Fax: +1 617/613-5000 • www.forrester.com

Table Of Contents

Executive Summary	3
Background And Methodology	3
Introduction: Why Software Change Management Matters.....	3
The Pains Of Ad Hoc Software Change	4
Business Pressures For Better Software Change Management	5
Technology Pressures For Better Software Change Management	5
Levels Of Software Change Management Maturity	5
Scope Of Software Change Management Processes.....	6
Enforcement Of Software Change Management Processes	7
Siloed IT Organizations Demand Coordinated Software Change Management	7
Geographical Distribution	8
Organizational Distribution	9
Technology Silos.....	9
Functional Silos.....	10
Recommendations	11
Assessing The Problem.....	11
Building The Business Case For Change.....	12
Conclusions	13

© 2006, Forrester Research, Inc. All rights reserved. Forrester, Forrester Wave, WholeView 2, Technographics, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. Forrester clients may make one attributed copy or slide of each figure contained herein. Additional reproduction is strictly prohibited. For additional reproduction rights and usage information, go to www.forrester.com. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.

Executive Summary

This study explores software change management — the definition and implementation of defined processes to control changes made to software. In this study, Forrester surveyed IT decision-makers to determine the state of software change management practices in enterprise IT organizations.

This study examines the obstacles to successful software change management, as well as the benefits enterprises can accrue by overcoming these obstacles.

While the majority of enterprises have implemented basic software change management processes, these processes fall down in the face of the increased complexity of today's business and IT environments. Truly effective and efficient software change management will require better coordination of change across geographic, organizational, functional, and technological silos in the IT organization.

Background And Methodology

To assess the state of software change management practices in enterprise IT organizations, Forrester conducted an anonymous telephone survey of 102 IT decision-makers at North American firms with more than \$500 million in annual revenues. This survey was conducted in the summer of 2006. Our respondents were decision-makers or influencers who are familiar with their IT organization's change management processes and who play the role of decision-maker or influencer when it comes to defining and implementing those processes. Sample respondent titles include director/VP of application development/systems development and director/VP of the project management office. The goal of this research, which was commissioned by Serena Software, was to determine the following:

- Level of adoption of defined change management processes.
- Level of adoption of change management tools.
- Complexity of changes and associated change management processes.
- Challenges and associated costs resulting from complex changes.
- Perceived benefits of better change management practices.

Introduction: Why Software Change Management Matters

Software change management is an essential discipline for enterprise IT organizations. In modern enterprises, software automates a wide variety of business processes. Changes made to software are, in effect, changes made to the business processes themselves. For example, the way an insurance company provides quotes for life insurance policies is encoded in its quote engine. Inadvertent changes to this quote engine can result in inadvertent changes in the types of quotes the firm provides. Software change thus requires careful management just as business process change does, as the two are inextricably linked.

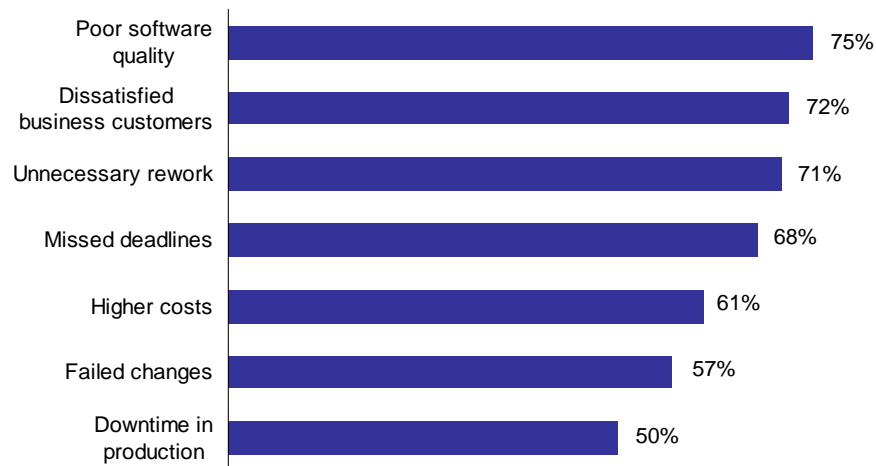
Without proper software change management, enterprises lack a full understanding of how software running in production automates their business processes. This includes management of changes to software in development, changes to software in production, and changes to associated artifacts like requirements, models, and test cases. This also includes management of both individual changes and the coordination of dependent changes.

The Pains Of Ad Hoc Software Change

The alternative to mature software change management is hardly an alternative at all. Software change management is a core practice for any IT organization, and IT shops that neglect this domain pay a high toll. We asked respondents that do not have defined change management processes in place for all of their software what types of pains they encounter as a result. Our respondents cited poor software quality (75%), dissatisfied business customers (72%), unnecessary rework (71%), missed deadlines (68%), higher costs (61%), failed changes (57%), and downtime in production (50%) as resultant pains (see Figure 1). Only 14% of respondents from companies that do not apply defined change management processes to all software rated the cumulative impact of these pains as minor.

Figure 1 Lack Of Software Change Management Processes Cause Problems In Critical Areas

“What types of pain does a lack of change management processes cause your company?”



Base: 28 North American \$500M+ companies, of 102 surveyed, that do not have defined change management processes for all software (multiple responses accepted)

Source: A commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006

Increasing business and technology pressures make ad hoc software change less and less of a possibility for today's enterprise IT organizations.

Business Pressures For Better Software Change Management

The decision-makers we surveyed believe in the benefits of software change management, and the current business climate puts them in a position where they must rely on it to help them meet the increasing demands of their business customers.

Even if technology were static, IT organizations would still be rapidly changing organizations, as the business world that they must align themselves with is highly dynamic. As the overall pace of business increases, so does demand from the business for IT to speed up its own internal operations. Changing competitive landscapes, mergers and acquisitions, increased proximity to the customer through use of Web 2.0 technologies, and even social computing all create pressure for IT to deliver software changes more rapidly.

And even while the speed of business operations increases, the demand for quality software remains as high as it ever was — if not higher. As enterprises come to rely more heavily on software to automate mission-critical processes, their tolerance for failed changes and production incidents decreases. This requires more disciplined change management practices.

These two pressures together result in a serious challenge for IT organizations: how to deliver software changes rapidly but with sufficient discipline to ensure quality. Forrester's enterprise clients frequently report difficulty implementing the right level of change management — enough to protect business interests without getting in the way of delivering new value to the business.

Technology Pressures For Better Software Change Management

Modern development languages, tools, and techniques put IT organizations in a position to meet business demands for speed by dramatically increasing the productivity of the development organization. But increased productivity simultaneously tempts IT shops to roll software into production without applying appropriate controls, which can ultimately waste as much as or more money than was saved through those same productivity increases.

Increased adoption of componentized software architectures is one of the main technical drivers for better change management. Examples of greater componentization include such familiar architectures as client/server, the Web, and cross-platform, as well as more recent innovations like service-oriented architecture. Enterprises that want to take advantage of the greater flexibility afforded by these architectures inevitably find that they must put controls in place to manage the risk that comes from having more moving software parts.

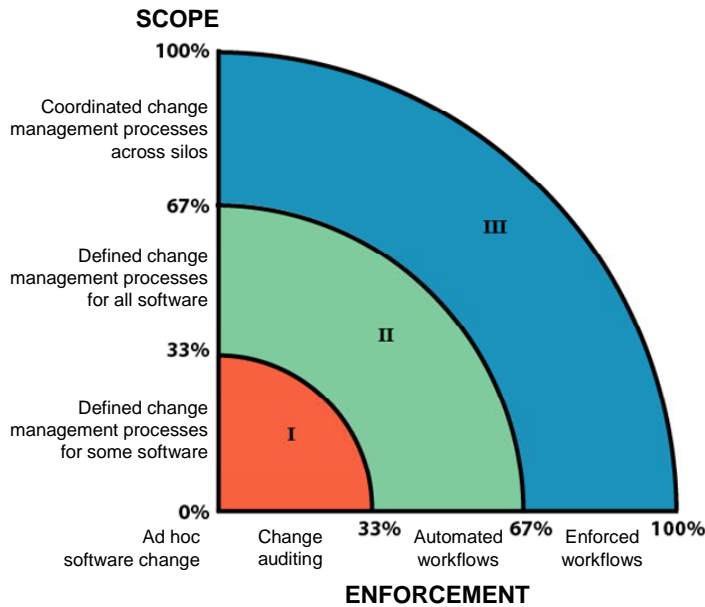
Today's challenges of managing change across platforms are significant though relatively well understood, but the challenges associated with software change management in an SOA environment are new or still in the future for most enterprises.¹ IT organizations that adopt SOA will find that the importance of mature change management grows exponentially. To start with, services are explicitly built for reuse, and changes to services thus have the potential to affect a large number of applications that consume them.² In addition, business services are embodiments of business processes and are therefore mission-critical software assets. Any defect in business services is likely a high-severity defect, and because the service is built for reuse, it's also a far-reaching defect.

Levels Of Software Change Management Maturity

Enterprise IT organizations thus face significant business and IT pressures to mature their software change management practices. There are two dimensions to software change management maturity: scope and enforcement (see Figure 2). The scope of the software change management

program expands from some software to all software and from management of individual changes to coordination of dependent changes. The enforcement of software change management practices starts with auditing during or after change implementation, then moves into automated workflows that guide users through the change management process, and finally leverages these workflows to prevent work from being performed outside of these defined processes. The Recommendations section of this report includes a self-diagnostic tool for IT organizations to use in assessing where they lie on each dimension. The scores produced by this tool can be used to plot where an enterprise falls in overall maturity of software change management practices.

Figure 2 A Map Of Software Change Management Maturity



Source: A commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006

Scope Of Software Change Management Processes

At the first level of software change management maturity, shops implement defined processes for managing changes to some software; at the second level, they implement such processes for all software. Most enterprises recognize the importance of software change management and have made the effort to define relevant processes. According to this study, 72% of the 102 enterprises have defined processes in place for managing changes to all software, and another 23% have defined processes in place for managing changes to some software. Public companies are even more likely to have defined change management processes, with 78% having done so for all software, and 18% having done so only for some software. This speaks to the impact of regulatory requirements like Sarbanes-Oxley on enterprise software change management practices.³

At the first and second levels of scope for software change management, the focus is on the management of individual changes. But as enterprises have come to rely more heavily on IT, the IT organization has grown in size and in complexity. As a result, IT organizations are riddled with organizational, geographic, technological, and functional silos. But because these silos are not fully

independent — and rightly so, since true independence would result in significant inefficiencies — changes inevitably span multiple silos.

The third level of scope for software change management involves the implementation of processes for coordinating change across the organization and across these silos. For example, change management is especially challenging — and critical — in the context of globally distributed development. Forrester defines globally distributed development as development spread across multiple geographies, organizations, and cultures, cultural distribution being nearly unavoidable in the event of geographical and organizational distribution. Our research shows that 45% of the enterprises have software development efforts that are spread across multiple teams *and* efforts that are spread across multiple geographic locations.

Enforcement Of Software Change Management Processes

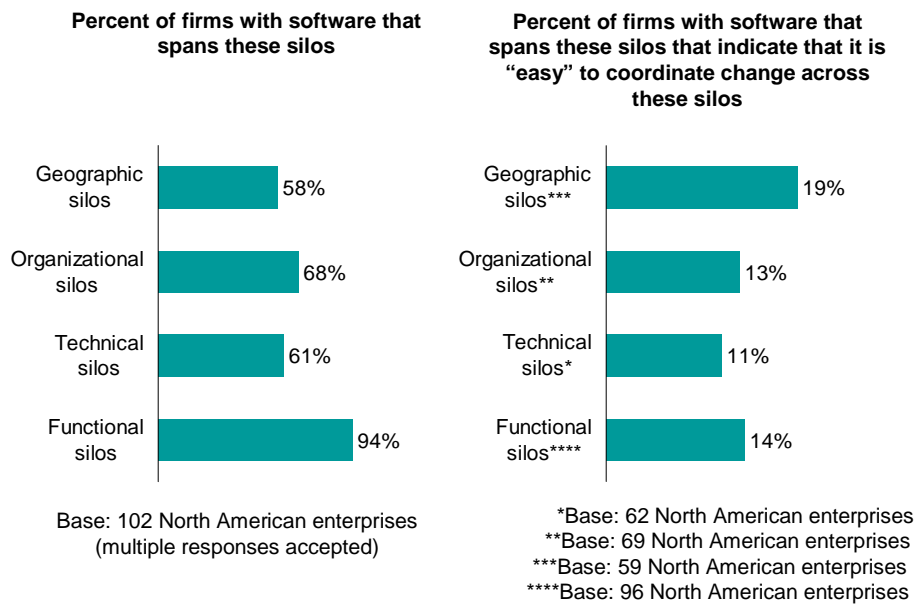
Broadening the scope of change management processes can help stop the bleeding, but simply defining broader processes isn't sufficient. Ensuring the successful implementation of change management processes — taking steps to ensure that these processes are followed — is where the rubber hits the road. The first step toward enforcement is the auditing of changes to ensure that they have been performed according to predefined processes. This can take place at any point in the change life cycle. Some IT shops perform audits at multiple points in the change life cycle; some shops perform audits only immediately prior to deployment as part of release management; and some shops perform audits only post-implementation. The first two types of audits are preventative measures that stop improper changes from being implemented, while the third type of audit is purely historical and is typically used to measure conformance with change management processes rather than to enforce conformance.

IT organizations turn to change management tools to increase both the efficiency and the effectiveness of their change management processes. Our study determined that 46% of North American enterprises with defined software change management processes use tools to automate all of these processes, and another 36% use tools to automate some of these processes. Some IT organizations rely on change management tools simply to guide the change process, but others use change management tools to enforce processes. For example, a change management tool can easily be used to prevent actions like checkout or promotion or deployment from being performed until the necessary steps in the process have taken place. The former represents the second level of maturity in enforcement of software change management processes; the latter represents the third level.

Siloed IT Organizations Demand Coordinated Software Change Management

But the complexity of today's business and IT environments makes efficient and effective software change management simultaneously more important and more difficult to achieve. Four primary factors complicate software change management: multiple geographies, multiple development teams, multiple technologies, and multiple roles. Virtually all (94%) of the firms we surveyed have change processes that span multiple roles. On top of this, approximately a third of enterprises (29%) are affected by software that spans multiple geographical, organizational, and technology silos; another third (31%) have software that spans two of these three silos; and another third (31%) have software spanning all three of these silos. Less than a fifth of the firms we surveyed indicated that it is easy to coordinate changes across any of these silos, across the board (see Figure 3).

Figure 3 Software Spans Multiple Silos, And Managing Change Across These Silos Is Not Easy



Source: A commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006

Geographical Distribution

Geographically distributed development is a reality for the majority of IT organizations. Our research shows that 58% percent of enterprises have software that is developed or maintained by firms in different geographic locations, and that's in addition to the firms that have multiple applications under development in multiple locations but have concentrated work on each application in a single place. The result is development organizations and even development efforts that are spread from Brussels to Boston to Bangalore.

Only 19% of firms indicated that it is easy to coordinate changes that span multiple geographies. For many of these shops, geographic distribution is a fait accompli no matter how difficult it might prove. Many firms report that the cost savings from labor arbitrage are so compelling that they cannot possibly be outweighed by increased cost in areas like software change management. And in other cases, the knowledge of applications and their histories that happens to reside at far-flung sites is so invaluable that it's worthwhile to maintain those sites for that reason alone. This is often the case for firms that have grown through mergers and acquisitions, as large enterprises have.

To mitigate the costs of geographic distribution, some firms attempt to create teams in each location and take pains to architect their software so that each collocated team can tackle components on its own. While this may permit the team to function as though it were collocated, it requires additional levels of coordination among the teams to overcome organizational distribution, and it leaves the geographical distribution of the overall development effort — if not the individual development team — intact.

Organizational Distribution

More than two-thirds of the firms we surveyed have multiple development teams working to enhance a single application. This includes both distribution of efforts among internal development teams, as well as distribution of efforts across internal and external or solely external teams. For example, governmental organizations commonly hire a prime contractor to build a software application for them; this prime contractor then hires subcontractors to tackle different components of the application. Coordinating dependent changes across these subcontractors, whose interests are often misaligned, can be very challenging. But organizational distribution is hard to avoid when building large software applications, whether or not third parties are engaged in the project. Firms that believe small teams to be more efficient than large teams typically choose to use five teams of 10 rather than one team of 50. However, inefficient processes for managing changes across these teams can eat into any productivity gains that come from using smaller teams.

According to our research, 68% of enterprises have software with components developed by different teams, whether internal and external. And of these firms, just 13% find it easy to coordinate the changes that those different teams are implementing. Foremost among these challenges were problems getting different teams working together effectively.

"Changes cause conflict between the teams, and hence the end product gets affected in terms of time of delivery." (*Director of information technology, life insurance company*)

"It's hard to get a common understanding between different teams and manage different team schedules." (*Director of information technology, manufacturing company*)

The firms we surveyed indicated that technical challenges obstructed change management across their development teams, too. Some respondents pointed to difficulty sharing access to a common code base, while others pointed to difficulties resulting from use of different source code management and change management systems.

"Our task is to coordinate version control between teams in specific time frames across the world." (*Divisional vice president, systems and applications, retailer*)

"We have different requirements for different source codes and mechanisms to control them. We also have to work on coordination among multiple teams working on the same source code." (*Manager, business systems, retailer*)

"We all have various tools to handle change management processes, and coordination among them is required." (*CIO, chemicals company*)

Technology Silos

Of the firms we surveyed, 94% run software on multiple platforms, and 52% run software on both distributed platforms like Windows, Unix, and Linux, as well as mainframe and midrange platforms like zSeries and iSeries. A full 61% of respondents run software that itself spans more than one of these platforms. Of these firms, only 11% find it easy to coordinate changes across these platforms. Virtually all of the challenges our interviewees cited were related to the management of dependencies between platforms and the synchronization of work on each platform.

"Changes have to be scheduled so that they're simultaneous." (*Director of information technology, pharmaceutical company*)

"Proper synchronization of software and byte code on different platforms is the hard part."
(*Application architect, transportation company*)

"If the change has to be backed out of any one environment, it has to be backed out of all environments it's running on."
(*Executive director, information technology, telecommunications company*)

"There's overall increased complexity that arises from having one app on multiple different platforms."
(*Director of information technology strategies and implementation services, nonprofit organization*)

Because different platforms are usually the responsibility of different organizations, management of changes across platforms requires management of changes across organizations, as well. These different departments frequently have different software change management processes, and interfaces between these processes are often undefined.

"This requires coordination between different departments and respective stakeholders."
(*Director of corporate technology, US hotel chain*)

"There is a change process in place for separate applications, but there is no cross communication and synchronization between applications."
(*Architect, enterprise systems, US educational institution*)

Functional Silos

As software has become more complex, specialization among those implementing changes to software has increased. The result? Functional silos in areas like business analysis, project management, architecture, development, database administration, testing, release management, support, operations, to name just a few. These are standard roles in enterprise IT organizations, and the emergence of functional silos built around them is a widespread problem. Functional silos pose an obstacle to coordination and communication. When every role is responsible for only one type of work, it's all too easy for a change to be improperly implemented, even though every individual has done his or her job. For example, when test cases aren't mapped to the appropriate requirements, it might not be until after that application has been deployed that serious deficiencies are discovered. Or if the wrong software changes are associated with a defect, the wrong patch can be deployed and defect resolution can be needlessly delayed. Ensuring that everyone's part adds up to the right whole is one of the major challenges facing IT organizations today.

To have confidence that software changes accurately implement business requirements, IT shops have to be able to identify the requirement behind each change, the actual work done to implement the change, and the steps taken to verify that these changes fulfilled the initial requirement. This requires careful change management both within each functional silo *and* across the full life cycle. Virtually everyone who touches software in an enterprise IT shop is required to follow a change management process; the firms we surveyed said that change management processes are applied to IT operations (89%), project managers (85%), development leads and developers (84%), QA and testers (73%), architects (67%), IT executives (67%), a release management group (60%), and a change control review board (64%). Just 14% of respondents find it easy to coordinate changes across all of these roles.

In organizations with less mature software change management practices, the processes these specialists follow are disjointed. Of those respondents with defined software change management processes, two-thirds have processes for managing changes to life-cycle artifacts like requirements, models, source code, documentation, and test cases. But half of the firms we surveyed use distinct

change management processes for different types of artifacts. This makes sense — the change life cycle for a test case is less complicated than the change life cycle for a requirement — but it also adds complexity and necessitates an additional level of coordination.

Recommendations

Enterprise IT organizations that recognize the value of coordinated software change management practices will want to assess their own organizations to determine the relative need for improvement. If such need exists — and for the vast majority of IT shops, it will — then the next step is to build a business case for meeting that need.

Assessing The Problem

Before a company turns its attention to the coordination of software change across silos, it should be sure that it has foundational practices in place. For example, are there defined software change management processes? And are these processes automated? Figure 4 is a self-diagnostic tool that readers can use to determine where on Forrester's software change management maturity scale their organization falls. The tool produces a score for scope of software change management processes and a score for enforcement of software change management processes. In our maturity model for software change management practices (Figure 2), the former represents the y axis and the latter represents the x axis.

Figure 4 Software Change Management Maturity Self-Diagnostic Tool

This two-part diagnostic tool can help you gauge the maturity of your change management processes and the degree to which they are enforced. Use these scores to determine where you fall on our map of software change management maturity.

	Yes	No
Do you have defined processes in place for managing changes to <i>some</i> software?	<input type="checkbox"/>	<input type="checkbox"/>
Do you have defined processes in place for managing changes to <i>all</i> software?	<input type="checkbox"/>	<input type="checkbox"/>
Do you have defined processes in place for managing dependent changes across silos?	<input type="checkbox"/>	<input type="checkbox"/>
Number of "yes" responses: ____ / 3 = ____%		
	Yes	No
Do you audit changes to ensure that they have followed the appropriate processes?	<input type="checkbox"/>	<input type="checkbox"/>
Do you use tools to automate change management processes as workflow?	<input type="checkbox"/>	<input type="checkbox"/>
Do those tools prevent work from being performed outside of the defined workflow?	<input type="checkbox"/>	<input type="checkbox"/>
Number of "yes" responses: ____ / 3 = ____%		

Source: A commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006

Once basic foundational practices are in place (e.g., defined processes are in place for managing changes to all software and these processes are enforced by automated workflow), then the IT

organization will be ready to turn its attention to the way in which it coordinates software changes across the silos of the IT organization.

After taking inventory and determining where disparate software change management processes exist, it makes sense to stop and ask whether there is real need for disparate processes in the first place. Change process standardization may be an option, but there are frequently valid reasons for variance in process (e.g., risk level), so proceed carefully. And if disparate software change management processes are truly necessary, integration will likely be the order of the day. Integration of process workflows, whether manual or automated, can help bring about coordination of software change that spans silos.

Building The Business Case For Change

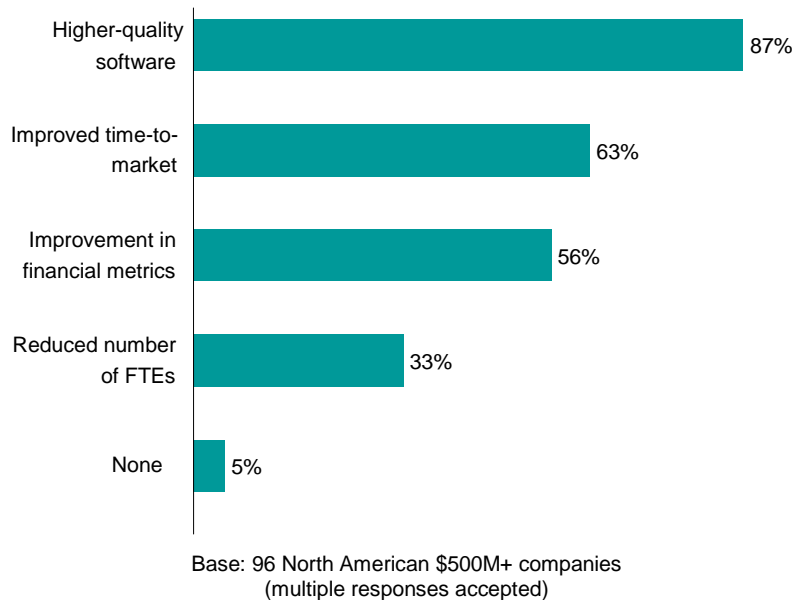
Identifying the need for better software change management is a small but vital part of the battle. Initiatives to put in place coordinated software change management practices simply cannot be driven from the ground up; the driving force behind such initiatives must be executives whose portfolios encompass multiple silos. The more far-reaching the portfolio of the sponsoring execs, the more they will be able to do to improve software change management practices at their firms. Building a strong business case for improvements to existing practices is absolutely necessary in order to secure the support of such executives.

The key to building the business case for change is to examine “as is” practices, determine potential “to be” practices, and calculate the expected costs and benefits for this transition. To guide efforts of this nature, we asked interviewees about the types of benefits they expect to accrue from better software change management practices.

Our findings? There is wide agreement on the positive impact of improvements in software change management practices. Respondents believe that better change management processes result in higher-quality software (87%), improved time-to-market (63%), improvement in financial metrics like ROI and TCO (56%), and reduced number of full-time equivalents (33%) (see Figure 5). They also cited benefits like “less rework” (manager of information technology, educational institution), “more predictability” (director of IT, aerospace company), “better understanding of the status of projects,” (VP of IT, professional services company), “better compliance” (director of IT, high-tech products company), “improved customer satisfaction” (VP of IT, construction materials company). The majority of IT operations professionals we surveyed believe that better software change management practices result in improved uptime in production.

Figure 5 Enterprises Believe That Better Software Change Management Has Far-Reaching Benefits

“What kind of benefits would you expect to accrue from improved change management?”



Source: A commissioned study conducted by Forrester Consulting on behalf of Serena Software, November 2006

The firms we surveyed are also very optimistic about the cost savings to be gained from improved automation of software change management processes. On average, respondents that do not have tools in place expect that they could reduce their overall IT budget by 5% by adopting tools to automate software change management workflow, with some respondents citing expected reductions as high as 25%. Your mileage may vary, of course. But the takeaway here is that the ROI of software change management is substantial. When building a business case for improvements in this area, be sure to take a broad view of the potential benefits to the IT organization and to the business.

Conclusions

For years, enterprise IT organizations have focused their attention on implementing defined change management processes and putting measures in place to ensure their enforcement. Internal and external auditors have driven home the importance of process enforcement, and the efficiencies to be gained from process automation are well understood. But definition and enforcement of change management processes is just a first step on the road to mature software change management practices.

Our interviews make it clear that the limited scope of most software change management efforts leaves much to be desired. The challenges associated with managing software changes across geographical, organizational, technological, and functional silos are significant. Change management across silos requires close collaboration of individuals who do not ordinarily collaborate. And siloed processes and process enforcement mechanisms — including but not limited to tools that automate these processes — stand in the way of this collaboration.

For most enterprises, the cost of this collaboration is high. But when this collaboration does not occur, the results can be dire. Better mechanisms for coordination of change across silos should be a priority for enterprise IT organizations. And the more heavily an enterprise relies on software, the more vital it is for that enterprise to implement more coordinated software change management practices.

ENDNOTES

¹ Forrester's Business Technographics® data shows that while 21% of North American enterprises currently have enterprise-level strategies in place to govern their use of SOA, another 18% are using SOA selectively, and an additional 21% are planning to pilot it in the next 12 months. Source: September 2006 Business Technographics Software And Services Study.

² When releasing new versions of services, IT organizations have two options: 1) maintain backward compatibility, or 2) maintain multiple versions of the service in production. In either case, careful change management of the service is required.

³ Section 404 of Sarbanes-Oxley, which affects all publicly traded companies in the United States, requires enterprises to have a precise understanding of how their production software operates. This requires enterprises to be able to trace from the original business request for change to the actual work done to implement the change and then steps taken to verify that these changes fulfilled the initial request.