

A Forrester Consulting Thought Leadership Paper Commissioned By Coverity

Software Integrity Risk Report

The Critical Link Between Business Risk And Development Risk

April 2011

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

Forrester Consulting
Making Leaders Successful Every Day

Table Of Contents

Executive Summary.....	2
Introduction And Survey Methodology.....	4
Software Risks Have A Direct Impact On Business Success.....	5
Increasing Pressure On Development	7
Traditional Testing Methods Are Falling Short.....	9
The Increasing Reliance On Third-Party Code Is Magnifying Risk.....	10
Key Recommendations.....	16
Appendix A: Methodology And Demographics.....	17
Appendix B: Endnotes	18

© 2011, Forrester Research, Inc. All rights reserved. Unauthorized reproduction is strictly prohibited. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. For additional information, go to www.forrester.com. [1-HM4APZ]

About Forrester Consulting

Forrester Consulting provides independent and objective research-based consulting to help leaders succeed in their organizations. Ranging in scope from a short strategy session to custom projects, Forrester's Consulting services connect you directly with research analysts who apply expert insight to your specific business challenges. For more information, visit www.forrester.com/consulting.

Executive Summary

In October 2010, Coverity commissioned Forrester Consulting to conduct a survey of 336 North American and European software development influencers. The purpose of the study is to examine the current practices with respect to managing software quality, security, and safety, and to identify key market trends and best practices.

Our study found a strong link between business risk and software risk; respondents reported that software defects have directly impacted customer satisfaction, revenue, brand image, and time-to-market. In addition, the No. 1 factor driving software quality initiatives is market leadership, which places development at the heart of delivering on business demands.

More than 65% of companies say software defects have the greatest impact on customer satisfaction.

As companies begin to understand that development risk directly impacts business risk, they are increasingly holding their developers accountable for the quality, safety, and security of the software they produce. Our study shows that business metrics are increasingly being built into developer performance reviews — more than half of the respondents said customer satisfaction was a factor in bonus, evaluation, and promotion decisions.

Increasingly, developers are expected to be experts in all areas of software defects, including quality, safety, and security, in addition to ensuring that the software meets functional requirements. Despite mounting pressures, developers do not utilize all the code analysis and testing technologies at their disposal. As a result, discoveries of code defects often happen late in the development life cycle or in production, which can cause significant post-development work; 44% of the survey respondents reported that defects found late by quality assurance (QA), security audit, or in the field were among the top issues likely to affect the success of a development project.

Making matters more challenging is the increasing reliance on third-party software. Almost every organization utilizes some form of third-party software, and many rely on software from multiple suppliers. In some industries, including mobile, the use of third-party code is extensive, with code coming from more than three to five different software suppliers. As such, third-party code can have a major impact on a company's bottom line and brand image. However, many companies do not practice effective management of quality- and security-related risks from third-party code today — our survey shows that organizations overall do not treat third-party code with the same rigor as in-house-developed code.

This study also found some important strategic shifts beginning to emerge at companies, based on the realization that development risks are tied directly with business risks.

- There is an increased awareness of software quality and security. Many organizations now realize that software defects do negatively impact business goals such as customer satisfaction, time-to-market, brand image, and company revenues.
- Organizations are increasingly holding developers accountable for software quality, security, and customer satisfaction.
- High-level business decision-makers and line-of-business (LOB) managers are demanding more visibility and insight into the quality and security of third-party-supplied code.

Going forward, we expect that the success of development projects will become more prominent in the minds of business leaders as companies continue to rely on software to achieve market leadership and operational excellence. We

also expect that businesses will demand more visibility into development risks, and that development teams will be increasingly measured on software as well as business metrics.

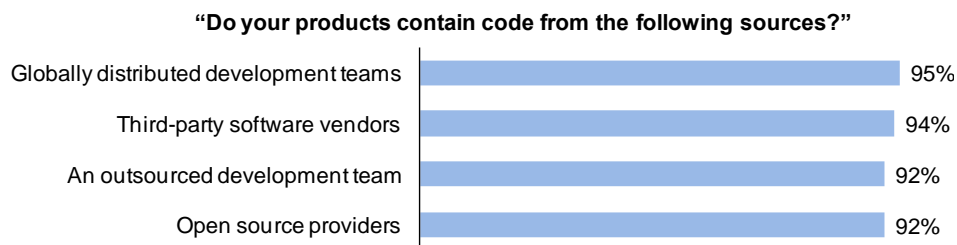
To keep pace with demands, development teams must deploy proactive technologies to shape the outcome of development projects. Tactically, this means performing analysis and testing earlier and catching more defects earlier in the development life cycle. Strategically, this means an overall shift in development practices: demanding quality and secure code from software suppliers and considering visibility into development risks as a business function central to a company's quest for market leadership and operational excellence.

Key Findings

Forrester's study yielded seven specific key findings:

- **Software defects have a direct impact on business goals.** Sixty-five percent of respondents say customer satisfaction is directly impacted by software defects, while 47% say software defects impact their product's time-to-market. Both customer satisfaction and time-to-market are business metrics, and as such, this is evidence that business risks are linked directly with development risks manifested in code defects.
- **Time-to-market and customer satisfaction drive firms to make developers more accountable for quality and security.** Nearly 50% of respondents say their firms are holding developers more accountable for quality and security issues today than a year ago. Many respondents cited software-related issues impacting time-to-market and customer satisfaction as the top two reasons driving this trend.
- **Organizations are heavily reliant on third-party-supplied code and geographically distributed teams.** Forty percent of our respondents say they work with more than five external software suppliers. Nearly everyone uses some form of third-party code in their software products, and 50% indicate that they use open source code regularly or extensively.

Figure 1
Providers Of Code



Base: 336 product development and IT professionals involved with software development (responses aggregated from a range of 1 to 5)

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

- **Half as many companies apply the same rigor to third-party code as they do for in-house-developed software.** Sixty-nine percent of respondents use automated testing during development, but only 44% demand it from their suppliers.¹ Sixty-eight percent perform manual code review on in-house-developed code, but only 35% apply the

same to supplied software. Similarly, 70% conduct risk/security/vulnerability assessments of in-house-developed software, but half as many — 35% — do the same for third-party code.

- **Awareness of quality and security issues drives demand for visibility into third-party code.** Nearly half of respondents say visibility into quality and security of third-party code is more important today than a year ago. Fifty-six percent report that this trend is driven by increased awareness of quality, safety, and security of supplied code; 47% also pointed to software-issue-induced product delays or recalls as another factor driving the importance of visibility into third-party code, and 32% stated problems with supplied code led to damage of their corporate brand.
- **Most companies use both business and technology metrics to measure developer performance.** Fifty-six percent of survey respondents say they use customer satisfaction to measure developers' performance. In addition, 51% use the number of critical software flaws left in released code as another metric for developer performance. In fact, a majority of companies consider defect avoidance (i.e., crash-causing defects, security and safety defects) to be a part of a developer's official job function.
- **There is no uniform way of testing practiced by development.** Currently, developers use unit testing as the most common testing method, our survey respondents say. But even that is only used regularly by little more than half of respondents overall. Still fewer practice static analysis, security testing, or manual code review regularly. As a result, 44% reported that software defects found late in development life cycle or in the field are most likely to impact the success of development projects.

Introduction And Survey Methodology

In October 2010, Coverity commissioned Forrester Consulting to conduct a study of North American and European software development influencers in order to better understand how organizations deal with software quality, security, and safety issues for in-house-developed as well as third-party-supplied software.

We surveyed 336 respondents across the US, Canada, the UK, France, and Germany. We drew respondents from companies producing software artifacts for internal consumption or commercial purposes (see Figure 2).² Respondents represent companies of different sizes, skewing toward large organizations; 50% are from firms with 5,000 or more employees, including 33% from the Global 2000. A small percentage, 3%, represents businesses with 100 to 500 employees. Respondents are technologists or product development managers directly involved with software development processes. Forty-two percent are software developers, 18% are development managers, and the rest are architects, QA personnel, security testing, project managers, product managers, and company executives (see Figure 3). We conducted the online survey between November and December 2010, asking 24 questions spanning development processes, security mechanisms, metrics, and organizational structures. To get more detail on respondent profiles, see Appendix A.

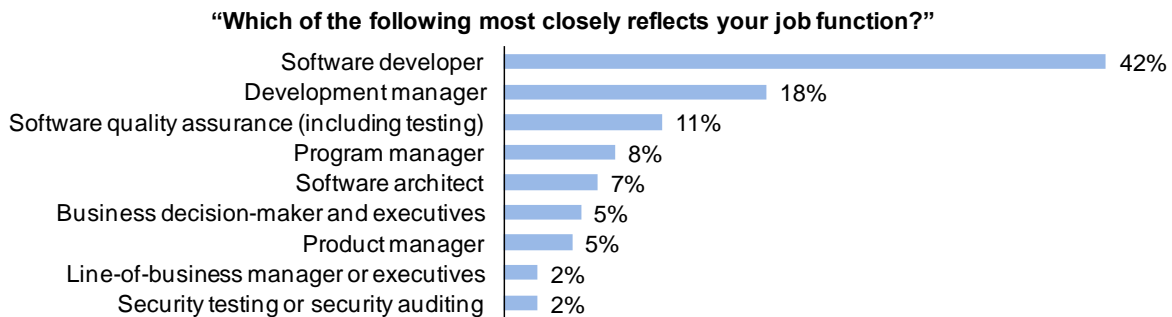
Figure 2
Respondents' Firms Produce Different Kinds Of Software



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Figure 3
Roles Of The Survey Respondents



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Software Risks Have A Direct Impact On Business Success

One of the main charters of this study is to examine the link between business success and development risks. To that end, we asked two questions. In the first question, we seek to understand whether there exists a link between business goals and software quality efforts. We asked the respondents to rank the top drivers behind their software quality initiatives. As depicted in Figure 4, our survey respondents listed market leadership and operational excellence, both clear business goals, as the top two drivers.

In the second question, we asked which business aspects — including customer satisfaction, brand image, and company revenues, etc. — are impacted by the presence of software defects. For this question, most respondents, 65% to be exact, say that “customer satisfaction” is impacted by software defects. Nearly 50% also indicated that “time-to-market” is another aspect impacted by software defects (see Figure 5).

These questions demonstrated a link between development success and business goals. It is business goals such as market leadership and operational excellence that inspired many of the software quality efforts. By the same token, business aspects such as customer satisfaction and time-to-market are affected by software risks.

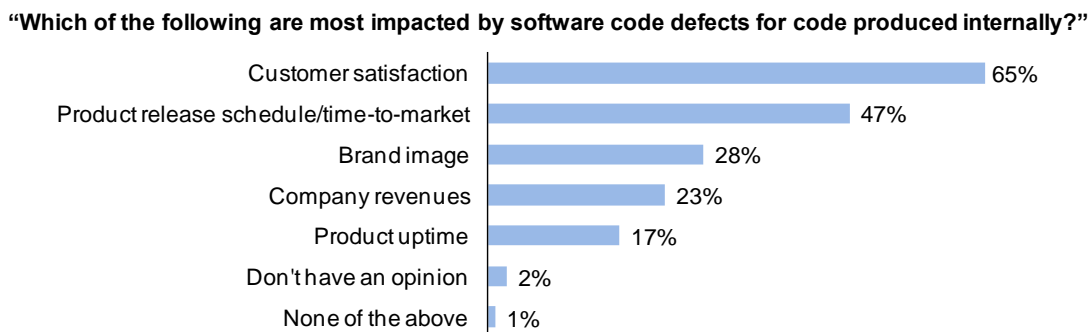
Figure 4
The Top Factors That Drive Quality-Related Initiatives



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Figure 5
Business Aspects Impacted By Software Defects



Base: 336 product development and IT professionals involved with software development

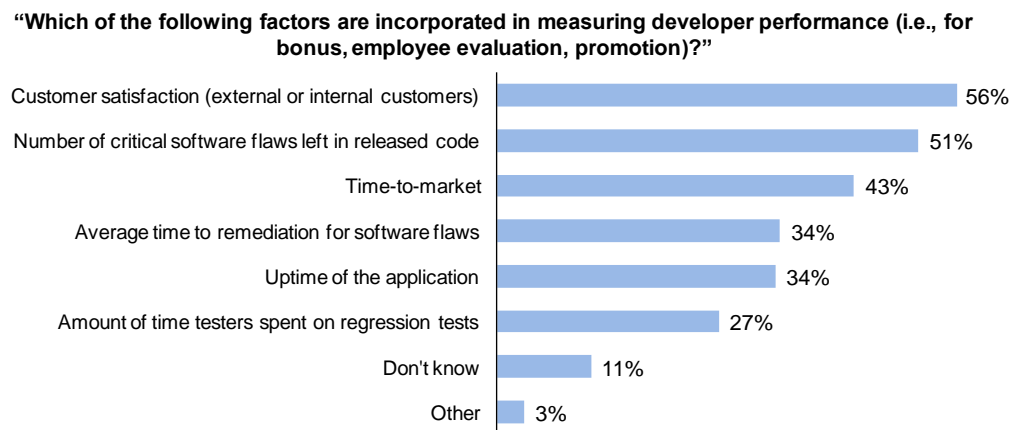
Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Increasing Pressure On Development

We found that many organizations today expect their developers to be responsible for software quality and security. In fact, a majority of surveyed companies told us that defect avoidance and producing secure code are part of developers' official job function.

When asked specifically about which metrics they use to measure a developer's performance and the success of a development project, customer satisfaction came up as the top metric in both cases. In both questions, however, more respondents chose quality metrics such as "number of critical software flaws left in released code" and "number of escalations due to defects" than those who chose "time-to-market" (see Figure 6 and see Figure 7).

Figure 6
Percentages Of Organizations That Include These Metrics To Measure A Developer's Performance

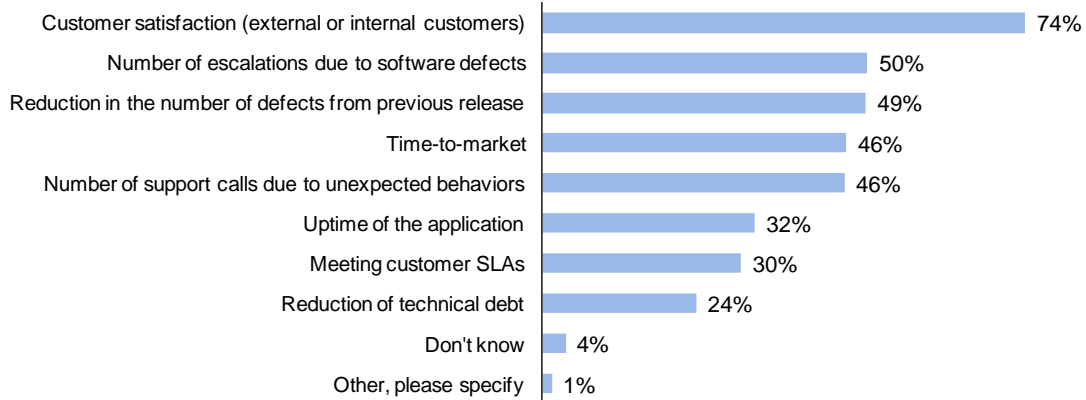


Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Figure 7
Metrics Used To Measure The Success Of A Development Project

“Which of the following factors are incorporated in measuring the success of your development projects?”



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

This is encouraging, as Forrester sees many development organizations under tremendous time-to-market pressure. At the end of the day, some may forego quality-related goals in favor of meeting delivery deadlines. This study tells a slightly different story: More organizations treat quality and security as priority concerns.

In a separate question, 74% of all survey participants told us that their developers are being held more accountable for quality- and security-related goals today than a year ago. When asked to share the reason behind this trend, these respondents, all 249 of them, cited product delays and adverse impact on customer satisfaction as the top two drivers (see Figure 8). Answers to this question provided concrete evidence that organizations see a direct link between code quality/security and business goals such as time-to-market and customer satisfaction. This also helps to validate why, in Figure 6 and Figure 7, more respondents chose quality metrics over time-to-market to measure the success of developers and development projects.

Figure 8
Reasons Why Developers Are Being Held More Accountable Today For Quality-Related Concerns



Base: 249 product development and IT professionals involved with software development

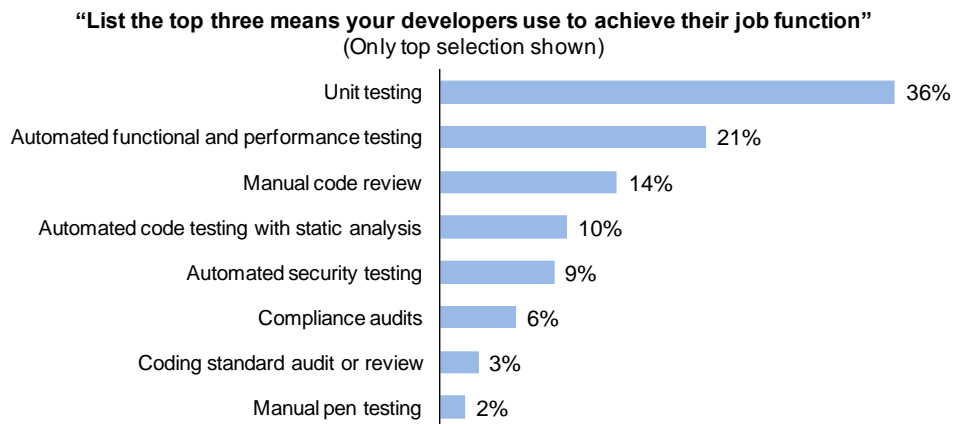
Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Overall, we see a trend that more pressure is on developers to be responsible for software quality and security. Development organizations are increasingly being measured with quality and security metrics, sometimes even above metrics such as time-to-market.

Traditional Testing Methods Are Falling Short

Although developers are under pressure to produce quality and secure code, they are not using all the technologies and tools at their disposal. When asked which code testing/analysis techniques their developers use to achieve quality and security assurance, unit testing came up as the most common choice; 36% say unit testing is the No. 1 means for testing that their developers employ. A mere 14% reported manual code review, and 10% say they commonly use static analysis (see Figure 9)

Figure 9
Top Code Analysis And Testing Techniques Practiced By Developers Today



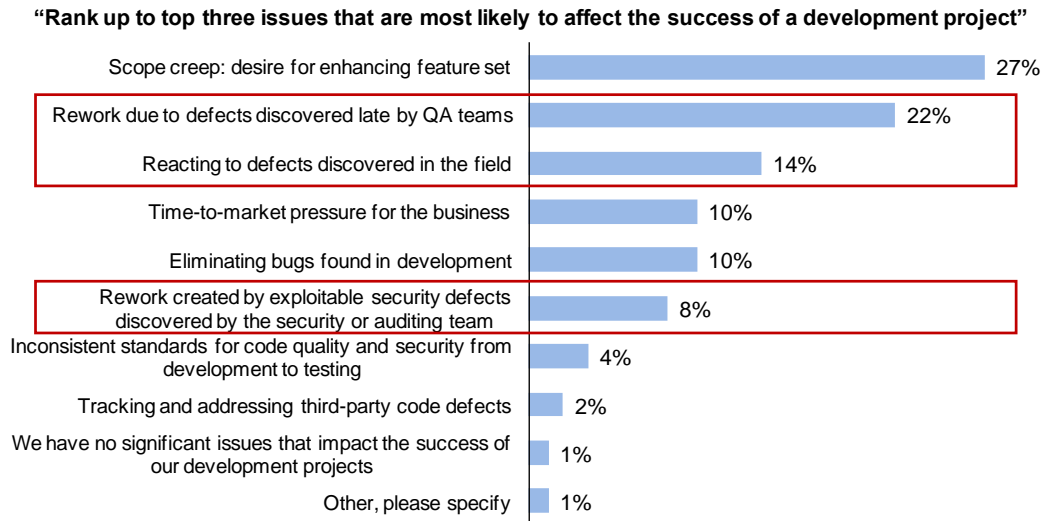
Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

As unit testing sees only a compartmentalized piece of code and often only covers functional aspects, it is not surprising that many defects are discovered late in the development life cycle. When we asked respondents to rank the top issues that affect the success of a development project, they listed defect-induced work late in the development life cycle as one of the top issues (see Figure 10). If software defects are not caught early in the life cycle, remediating these defects may require significant amount of effort on the part of developers and testers. As developers spend more time chasing down bugs post-development, testers will have to spend more time running regression tests.

Figure 10

Top Issues That Are Likely To Affect The Success Of A Development Project



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

The Increasing Reliance On Third-Party Code Is Magnifying Risk

Organizations may find that even if they address the risk in their own development processes, they’ve only solved half the problem. That’s because more than 90% of respondents indicated that they use some form of third-party-supplied code from commercial vendors, an outsourced team, or the open source community. In this study, third-party-supplied code is defined as commercial code, outsourced software, and open source code.

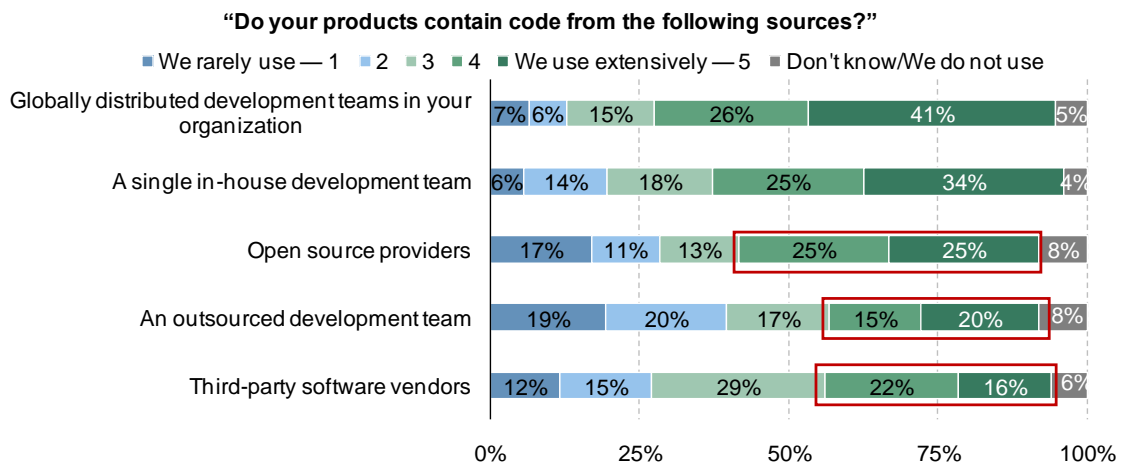
Though in-house development is still, by far, the predominant development method for our respondents, with 96% using it with varying frequency, a sizable percentage say they use software from third-party sources extensively (see Figure 11). Interestingly, open source code appears to be used more often than commercial software and development outsourcing, with 92% using open source overall and 50% using it regularly or extensively.

The use of third-party code is even more prolific in small companies. One hundred percent of the respondent companies that have 100 to 500 employees reported using all three forms of third-party-supplied code. Interesting vertical differences also exist. In our sample, for instance, government and financial services use very limited software outsourcing, while those respondents who develop software for the mobile industry reported extensive and liberal use of outsourcing.³

In addition, many respondents work with multiple software suppliers. Twenty-seven percent of the respondents have more than 10 suppliers, while 40% say they work with more than five suppliers. With this many sources to obtain software, effective risk assessment and management for third-party code can be a challenge.

We also found that many organizations are not treating supplied code with the same level of rigor with which they treat in-house-developed applications. Shown in Figure 12, 75% of surveyed organizations use automated QA testing, while only 51% perform such tests for supplied code. Similarly, 68% of all organizations perform manual code review on in-house-developed code, while only 35% do the same for supplied code. These data suggest that quality issues and security vulnerabilities within third-party-supplied code have a higher probability of being left untreated. Given the prevalence of third-party code, this is of serious concern.

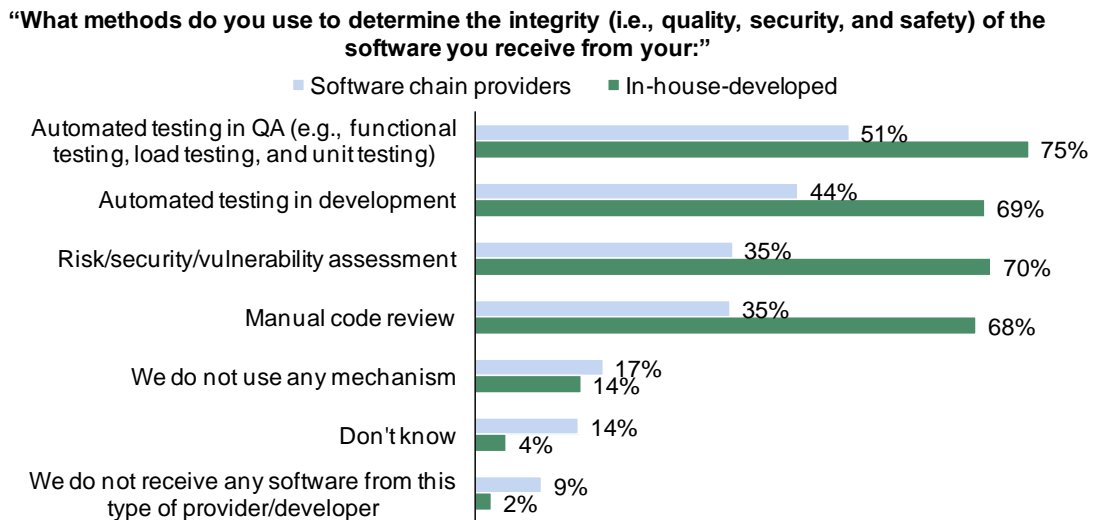
Figure 11
 Sources Of Software And How Extensively Organizations Use These Sources



Base: 336 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Figure 12
Methods Used To Determine The Integrity Of In-House Versus Provider Software



Base: 336 product development and IT professionals involved with software development

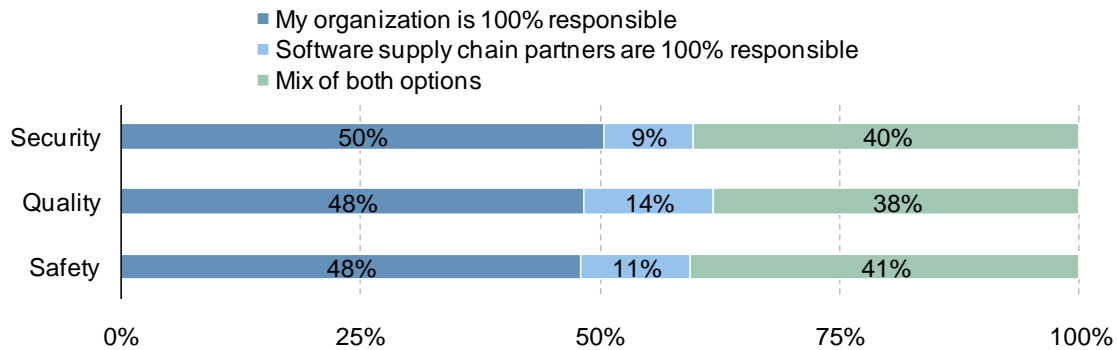
Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Today, The Buyer/Consumer Side Takes Up Most Of The Risk

Once the third-party code is integrated into a firm’s working product, our respondents told us that they, the buyer/consumer side, would be primarily responsible for the security, quality, and safety aspects of the software henceforth (see Figure 13). More specifically, in nearly one out of every two cases, the buyer side is 100% responsible for these quality-related issues, while in contrast, in only one out of 10 cases is the supplier held 100% responsible. This suggests a skewed risk-to-responsibility culture: The software producer/suppliers are really best suited to deal with quality-related issues with the software, yet they are often not the ones who are held responsible.

Figure 13
Responsibility Delineation After Code Integration

“Once the code is integrated into your working product, who is accountable for the following aspects of the software?”



Base: 333 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

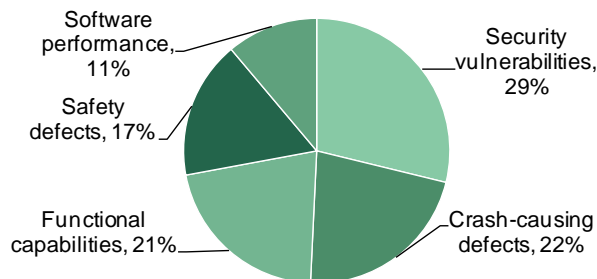
Organizations Increasingly Demand Insight Into Quality And Security Of Supplied Code

We asked the respondents to rank the top issues for which they’d like to have visibility, with respect to third-party-supplied code. The issues range from functional capabilities, performance, and defects, to security vulnerabilities. The respondents ranked security vulnerabilities as the top issue, outranking functional capabilities. Crash-causing defects is the second-highest-ranked issue, also outranking functional concerns (see Figure 14).

Different applications are concerned about slightly different issues with third-party code. Table 1 shows the breakdown along the application categories. It is not surprising that the top-ranked concern for web applications is security vulnerability, while it is safety defects for embedded software producers.

Figure 14
Top-Ranked Issues With Respect To Third-Party Code For Which Organizations Like To Gain Visibility

“How important is it to you to have visibility into the following issues of software supplied by a third party?”



Base: 333 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

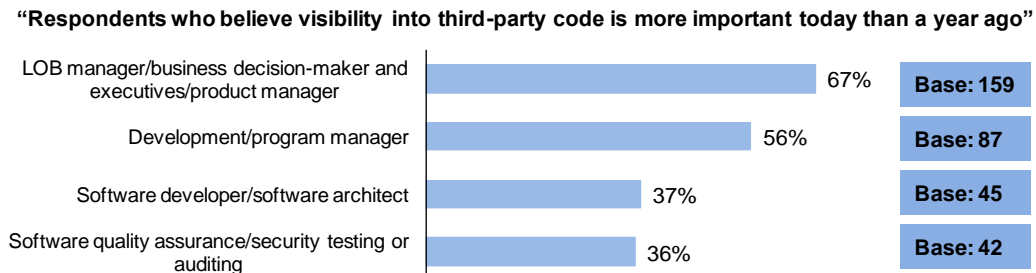
Table 1
Top-Ranked Issues For Different Application Producers

Top-ranked issues	Respondents who write web applications (n = 205)	Respondents who write embedded applications (n = 159)	Respondents who write consumer software (n = 186)	Respondents who write B2B enterprise software (n = 164)	Respondents who write cloud applications (n = 151)
No. 1	Security vulnerabilities (31%)	Security vulnerabilities (28%)	Security vulnerabilities (27%)	Security vulnerabilities (30%)	Security vulnerabilities (30%)
No. 2	Functional capabilities (22%)	Safety defects (27%)	Safety defects (23%)	Safety defects (21%)	Crash-causing defects (22%)
No. 3	Crash-causing defects (20%)	Crash-causing defects (23%)	Crash-causing defects (21%)	Functional capabilities (21%)	Safety defects (21%)

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

In addition, 46% of the respondents told us that the ability to gain insight into the quality and security issues of the third-party code is more important today than it was two years ago. More specifically, those who assume managerial roles for software development — including development managers, product managers, LOB managers, and business decision-makers — expressed a stronger desire to have such insight into third-party code than the other more tactical roles (see Figure 15).

Figure 15
A Role-Specific View Of Those Who Think Visibility Into Third-Party Code Is More Important Today

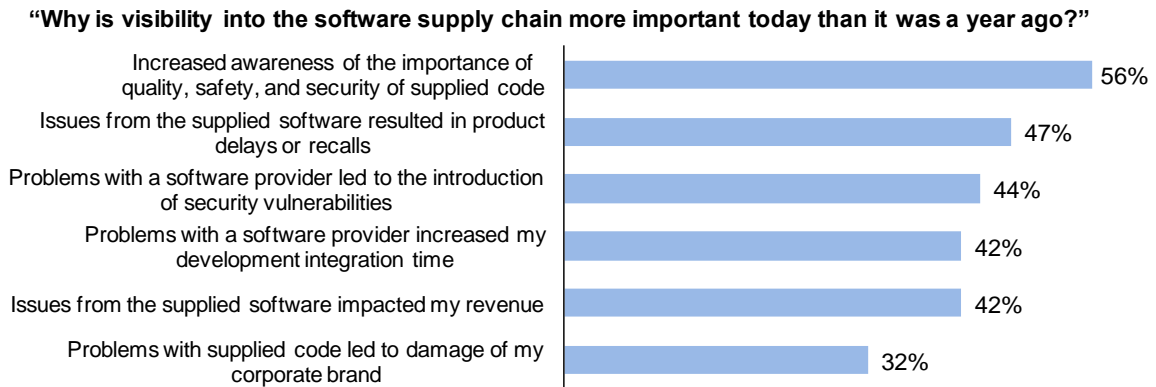


Base: 333 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

When asked about the factors that drive the desire to obtain more visibility in supplied code, respondents cited increased awareness for quality, safety, and security issues, as well as past problems with supplied code as the top impetus. Past problems with supplied code had a direct impact on business results including: product delays or recalls, impact to revenue, and damage to the corporate brand (see Figure 16).

Figure 16
Drivers Behind Why Organizations Want More Visibility Into Quality Issues Of Third-Party Code



Base: 152 product development and IT professionals involved with software development

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

KEY RECOMMENDATIONS

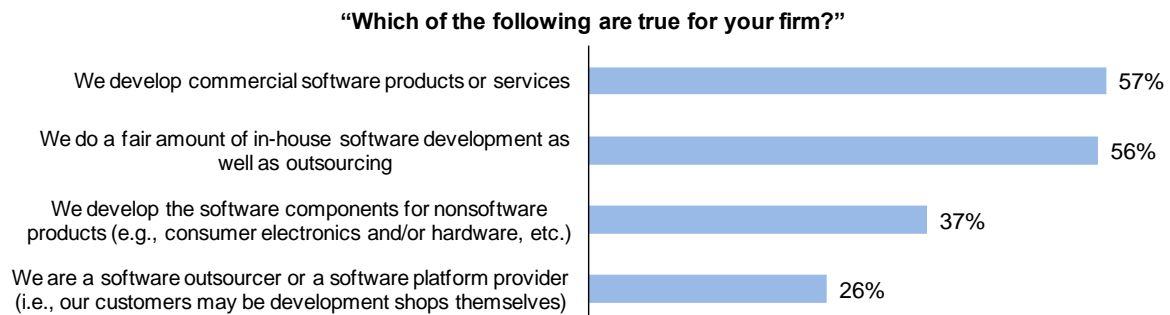
This study uncovered a number of findings, which collectively point to these best practices for development organizations:

- **View software development in business terms.** Because development risks directly impact business goals, business decision-makers and LOB managers must rethink development practices and mandate a more mature process that is capable of delivering quality and security assurance. A shortsighted practice that focuses on shallow metrics such as time-to-market can ultimately hurt the business bottom line and longer-term time-to-market.
- **Bake quality, security, and business metrics into performance evaluation of developers and projects.** Improving code quality, security, and safety requires proper coding practices, technologies, as well as success metrics. If quality and security metrics are not part of the performance evaluation, no amount of coding practices and standards will matter. Because business outcomes and development outcomes are so tightly linked, business-related metrics such as customer satisfaction indicators should also be incorporated into performance evaluations. You need to give developers incentives to follow the proper practices and ensure the highest-quality code.
- **Shift testing upstream into development.** Manual approaches to testing and homegrown tools are often not sufficient to mitigate development (and therefore business) risks. Organizations must push software testing upstream into development, utilizing technologies such as architecture modeling, static analysis, and dynamic testing to detect and resolve defects early on in the development life cycle.
- **Apply the same rigor to third-party-supplied software as in-house-developed ones.** The prevalence of third-party code in development projects means that they are critical to the success of the projects. Whenever possible, you should apply the same rigor — including automated analysis, manual code reviews, or penetration testing — to third-party code as you would to in-house-developed software.
- **Demand mature development practices of your suppliers.** When you work with a software supplier, you need to make sure that they use adequately mature development practices and produce evidence that quality, security, and safety requirements are met. Consider including that as an explicit requirement in your RFPs and also as an item for review in the ongoing vendor management process. It also means building remediation conditions in your code acceptance step; if the third-party code fails the test, you send it back until it is no longer the case.

Appendix A: Methodology And Demographics

In this study, Forrester conducted an online survey of 336 product managers and software development influencers and decision-makers in North America and EMEA. Survey participants included leads in engineering, development, product management, and product strategies. The study began in October 2010 and was completed in December 2010.

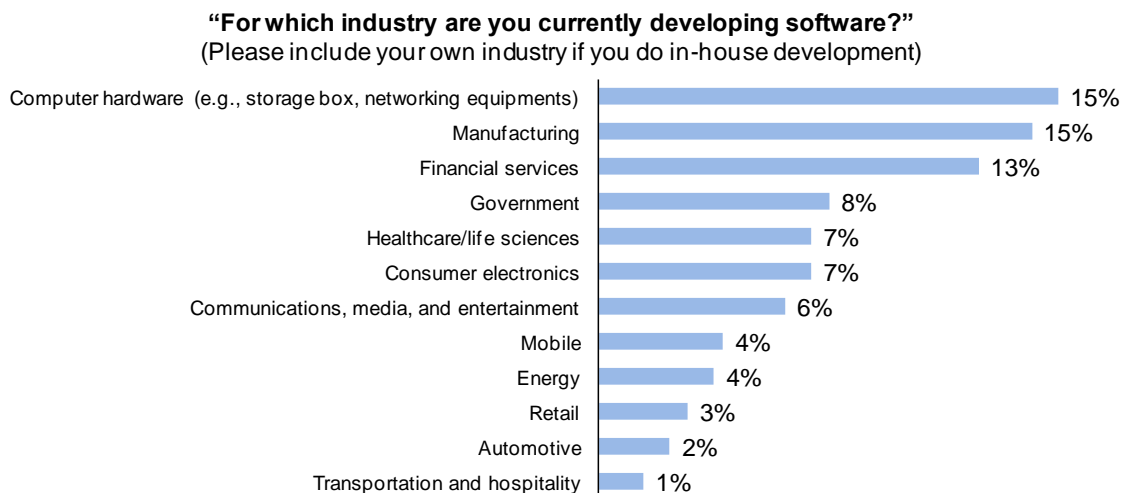
Figure A
Development Profile



Base: 336 North American software development influencers and decision-makers

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

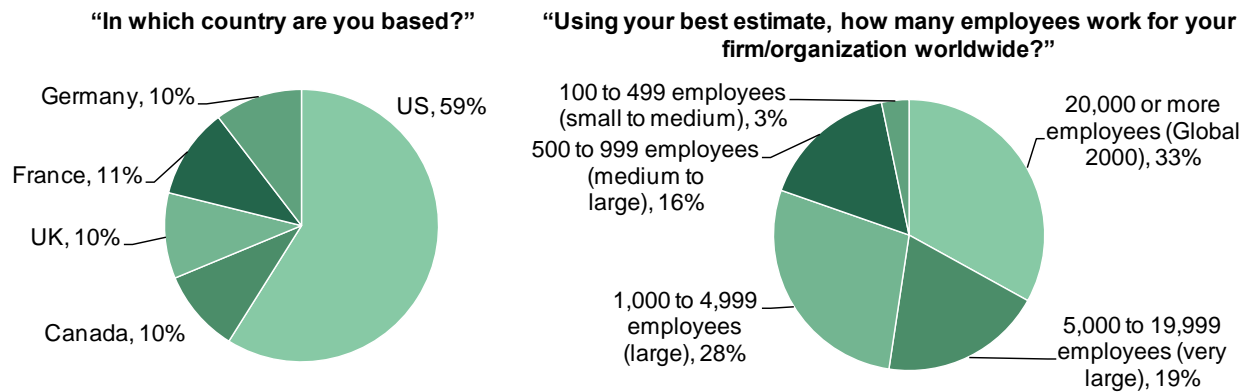
Figure B
Industry



Base: 336 North American software development influencers and decision-makers

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Figure C
Respondent Geography And Organizational Size



Base: 336 North American software development influencers and decision-makers

Source: A commissioned study conducted by Forrester Consulting on behalf of Coverity, December 2010

Appendix B: Endnotes

¹ It's worth noting that while overall data points to the lack of rigor applied for third-party code as opposed to in-house-developed software, UK and French respondents are exceptions; respondents in these countries report similar testing/assessment processes for third-party code as compared with in-house-developed code.

² Thirty-seven percent of our respondents produce software components for nonsoftware products such as consumer electronics, computer hardware, network equipments, etc.

³ Due to space limitation, we did not graph this particular point. Our respondents were developing software for these industry verticals: computer hardware, manufacturing, financial services, government, healthcare /life sciences, consumer electronics, communications/entertainment, mobile, energy, retail, automotive, and transportation/hospitality. Overall, 20% reported that they use software outsourcing extensively. Only three out of 26 of those who produce software for government agencies said they use outsourcing extensively. Healthcare came in at two out of 24. Financial services usage was 21%. In contrast, more than half (nine out of 14) of those who develop mobile software reported extensive use of software outsourcing.