



## TECHNOLOGY SPOTLIGHT

# The Service Provider Conundrum: Optimizing Quality Analysis for Enhanced Business and Client Value

Sponsored by: SonarSource

Melinda-Carol Ballou                      Jennifer Thomson  
July 2015

## INTRODUCTION: UNDERSTANDING THE IMPACT OF QUALITY ANALYSIS GAPS

---

The shift to a digital world, the impact of digital transformation, and demand for continuous deployment across technology platforms places huge pressure on IT organizations and the service providers that support them as they address dynamically evolving business needs. Time to market of high-quality applications becomes critical, but delivering software releases and developing new customer-facing applications quickly is a growing challenge. This is particularly the case for large, multinational organizations and service providers that are managing engagements. They must contend with a complex web of changing, multimodal technologies combined with legacy systems and resources across thousands of users that are geographically distributed. The creation of high-quality software in this context is particularly challenging.

IDC finds that this shift to the digital and mobile world fuels reliance on and demand for external application testing services. Based on IDC's *U.S. Application Outsourcing Survey* in 4Q13 and the *European QA Survey*, we find that around 60% of large organizations use service providers for application testing and a further 25% intend to use service providers for application testing in the next 24 months. Developing and delivering software releases and new customer-facing applications on time is a growing challenge and requires domain expertise across areas where staffing remains limited. Add to this the unrelenting pressures within the enterprise to beat the competition based on quality while looking to keep operational costs low and it becomes clear why many organizations seek third-party testing services instead of building capabilities themselves.

These factors combine to create high levels of software project complexity. Enterprise adoption of geographically distributed development using internal and external resources with system integrators (SIs) has increased significantly, along with the use of a variety of development tools (including open source), a range of languages, and agile processes. As a result, the need for proactive, common access to information about software problems with quantitative and qualitative metrics across disparate teams has never been greater both for end-user organizations and for service providers.

And in this era of digital consumerization and mobility, projects become both more challenging and critical to manage. Why? Enterprise service projects in the context of fierce global pressures with unprecedented levels and rates of technology change demand that departments work together, encompassing providers, business stakeholders and executives across IT, development, quality, and operations in order to survive competitively. But how can they get common, timely information about the code that's being created?

Visibility into code quality becomes critical as a metric to help improve behavior, to produce better software, and to map teams to suitable projects given experience levels and execution. Service providers and internal staff alike need to "up their game" with regards to software creation,

appropriate quality, and addressing defect backlogs. Service providers face the dual challenge of honing their own global development practices at the same time as bringing better quality, lower cost services to clients. "Technical debt" can only be identified and addressed if it is visible, which can be enabled via appropriate automation and code analysis tools and process strategies for adoption and analysis to understand the impact.

The bottom line is that service providers are accepting more risk as clients ask partners to "put more skin in the game," deliver business value, and ultimately take more ownership of the business risk. IDC research finds that many service providers are tasked with answering the following:

- How often are software changes being deployed back to clients and how rigorously have they been tested or analyzed (if at all)?
- How many people are involved in software development and how well are they managing code quality? How can we improve team productivity at a global scale?
- How should legacy application challenges be addressed to ensure competitive rates and reduce long-term maintenance costs?
- How can client confidence in software delivery be increased and how can quality levels of software be proved and demonstrated?
- How can we execute and provide differentiation in engagements?
- How can service organizations successfully reduce technical debt?

This IDC Technology Spotlight will discuss strategies for service providers to improve approaches to software analysis, defect management, security, and metrics to gain business and IT benefits via proactive visibility. Those who don't know history are doomed to repeat it; that adage also applies to ineffective approaches to software creation, quality, and defect mitigation and resolution.

## THE CHANGING QUALITY PERSPECTIVE

---

As businesses become more dependent on technology and on software to deliver core offerings, the job of the service providers in support of CIOs is shifting. Budgets are stagnant and yet the business expects IT to be able to deliver new business value more quickly than before in highly complex environments with increased expectations of quality. In addition, there is an expectation that IT becomes more aware of emerging business objectives and align IT priorities to match. A key challenge is how to enable faster delivery of better quality applications that address business and customer needs. Service providers, IT, and CIOs must move beyond improving IT efficiencies to enable business innovation and provide superior customer experience with continuous deployment and improved software practices. And a basic building block for doing that is code analysis and management.

Lack of visibility into code quality and resulting code that is rife with problems is both more obvious and more debilitating to businesses now because of the visibility and exposure demanded by mobile and other customer-facing applications. This has become a core issue for executives and those at the code creation level of organizations.

Developing and delivering high-quality software releases and new customer-facing applications on time is a growing challenge, particularly for large enterprises that must contend with a complex web of modern technology combined with legacy systems and resources. Organizations must address the needs of the internal business – the extension and availability of enterprise apps to mobile/social/cloud platforms – as well as satisfying client/consumer demand for the availability of innovative applications. Yet how is that possible without providing insight into problems as they are inadvertently created?

What can be done to improve overall software quality to reduce issues and risk while at the same time accelerating release cycles to bring new services and products to market faster? This demands effective and earlier management of software vulnerabilities and defects to enable proactive quality strategies and cut costs. Yet how can poor existing behaviors be mitigated to best accomplish those goals?

## MANAGING AND ENHANCING LARGE-SCALE SOFTWARE ENVIRONMENTS

---

### Challenges in Software Development

As service providers support executives to evolve business-critical applications, strong coordination of quality management at the earliest software life cycle phases through to operations can help lead to successful, continuous deployments and corporate and IT productivity. Organizations must make this transition by observing current challenges and leveraging strategies based on those issues to create the impetus for change.

IDC identifies the following as the top challenges facing software developers and organizations that depend on speedy, innovative software creation for competitive success:

- The business has ever-increasing needs both for upgrades and fixes as well as new customer-facing apps. IT and development teams must shift approaches to code and quality analysis to produce a higher level of confidence in the software delivered (with fewer defects).
- Enable faster time to market – at the right cost and quality, and with the right level of risk and the appropriate resource allocation – to shift the view that software development, code analysis, and testing are merely a "cost center" to being a core "business value enabler."
- Limit business risk and technical debt – help ensure the delivery of high-quality "secure" applications that work well "first time" through proactive, iterative code examinations, and common visibility for software practitioners and management.
- Shift from defect detection (often too late) to proactive defect prevention and upfront defect identification with common analysis and a "source of truth" across teams and executives.
- Flexible, contextualized dashboards can enable teams to view data that is relevant to them and can empower individual responsibility and collective collaboration across groups.
- Transparency is key to understanding quality, code completeness, and testing activities that are underway – coordinate with iterative build management and set appropriate thresholds.
- Ever-growing needs for governance and regulatory compliance are factors that affect the market and put pressure on companies to automate code assessment capabilities; this can help to drive workflow improvement, traceability, reporting, and metrics.
- Where there are limited or no unit tests and where there is little idea about whether the current design allows for the quick addition of new features or the ability to make improvements, apprehension about refactoring can straightjacket organizations and act as a spur to change.
- When bugs are discovered late in the development life cycle causing "red alert" situations and when releases become "dragons" that QA and customer support have to fight before software can be deployed to customers, that negative impact can also motivate behavioral shifts.
- Leverage code analytics information to provide the opportunity to improve behavior through transparency and iterative change (do not use this information as a "wall of shame" to punish teams or individuals).

- Service providers, IT, quality, operations, and business teams should unite and use these challenges as a jumping off point and as an impetus to shift poor existing software development habits and behavior. Establish effective software analysis and inspections, create coding rules and boundaries for builds and releases that require certain quality levels, and leverage metrics to improve quality strategy as well as the business outcomes that are reliant on high-performing, innovative software.

## Enhancing Application Value & Decreasing Software Development Risk

Successful service organizations are moving quality analysis up front in the process and making it iterative as part of the overall software development life cycle. Development and test worlds and business approaches as well are evolving toward agile models – continuous value generation/continuous improvement/continuous inspection are essential strategies to improve quality. These are some of the key strategies that IDC is observing for both end-user customers and service providers:

- Traditional approaches to code quality typically fail when testing becomes a check list item at the end of the development process with the QA team running a tool and going back to development with a list of actions that they need to take before the software goes to production. This is too late and costs too much – the product is now late or behind schedule and over budget etc. So what can happen all too frequently is that management signs off to move the project along with no quality gates for development.
- Collecting appropriate metrics with code analysis from projects starting at project inception is critical. Without the right metrics (or any metrics) the code base may be deteriorating without anyone noticing, and/or they only notice when the technical debt has reached a certain level where it's too expensive to address, given the time and budget constraints. Collecting code metrics continuously can provide visibility and give the team the advantage of keeping the technical debt of the code base under control.
- Increased certainty that the right code is being fixed is also core. Frequently, teams start refactoring because they believe that the code base is bad in terms of performance, brittleness, instability, or difficulty to maintain and/or to extend. But without the right contextual analysis it is impossible to detect which part of the code base is responsible for the issues encountered. So changes may be applied to the wrong code, or the right code gets refactored in the wrong way, or only part of the problem gets fixed. This is an area where metrics and tools can help by identifying the parts of the code that are causing the problems.

## Shifting from Technology Service Provider to Trusted Business Partner

More projects means more business requests which means departments (IT, QA, business) must work together to reduce the burden. IDC finds on average that organizations spend approximately 28% of business change budgets on change requests, and that a significant number of organizations are spending in excess of 50% of project budgets on testing. Results underline that operational improvements are a must, particularly as organizations now deal with a higher volume of projects.

Reliance on external providers is not just to help internal organizations deliver on a greater number of projects at reduced cost, but to ensure higher levels of quality and reduce business risk. In fact, IDC finds that the top benefits of leveraging third-party services are focused on improving business and customer satisfaction as well as software quality and agility.

Since the lines of business (LOBs) rely on software more than ever that is deployed across environments that are increasingly complex, they require specialized domain expertise. Adaptive and dependable software QA testing services help to deliver and sustain business value and success. Testing service providers must facilitate that transition with process and organizational

change coupled with effective software analytics and automated technology tools as part of their portfolio (from open source to enterprise "for purchase" options).

One thing is certain – although demand for external services picks up, in light of intense competitive pressure value propositions (combined with a solid services portfolio) must be strong and convincing, must speak to business-specific challenges, and must demonstrate value creation and continuous improvement. Moreover, service providers must execute and over exceed on requirements and expectations from the outset. IDC research finds that client tolerance for tardiness, error, or low quality of service is at breaking point. Close to a quarter of organizations that switch service provider or terminate a contract do so due to poor quality of service.

Client organizations are placing more demands on the service provider to accept more risk and provide greater visibility into the quality of service delivered. Put simply, end-user organizations want to know that the quality of the software (either maintained or developed) is of high quality and is being continuously enhanced – therefore the level of business risk associated with an app going live is kept to a minimum.

In this context, it is extremely important for service providers that provide both development and testing services to assure the client that there is a clear separation between the development and the testing parts of the business – in other words they are able to prove independence of these functions. This is necessary to generate confidence that there are the necessary quality gates in place to ensure quality software development. This requires that service providers are able to provide the client as well as internal teams with insight and control over the quality process.

Confidence in software delivery, a reduction in technical debt, a clear view of the pipeline and the software delivery process enabled by visibility and transparency into the code that's been created – all of these factors enable a collaborative working relationship between service providers and customers. Clients want service providers with which they can work closely and that have the appropriate domain expertise to execute well in these complex environments to ultimately enable innovation.

Differentiated services that deliver business value are key to winning future engagements. Service providers should position themselves and demonstrate that they are the essential "quality gate" between a client's IT and business organization. In other words they are the essential layer that sits between these two functions charged with weeding out the rubbish before the business gets involved (again). This should allow the business to have more confidence in the applications that IT is pushing out, resulting in higher business satisfaction and helping IT position itself as a business value enabler versus cost center.

## Ensuring Continuous Software Quality is Required to Succeed

Service organizations (and, increasingly, end-user companies) must run software development as a business. Part of doing that is managing the long-range maintainability of software – i.e., what you develop today you need to maintain tomorrow. Being proactive about the architecture and the quality of what is being created is key to managing long-term expenditures for software maintenance; it is more expensive to keep faulty, poorly structured software alive.

In this context, application maintenance teams also need visibility into software to be able to better maintain code to reduce costs, to enable better quality, and to improve customer responsiveness and ROI. Before any code is sent to the customers, establishing a consistent process for code analysis can help ensure long-term improvements for companies that must run software development effectively.

### SonarSource: From Open Source to Production Level Code Analysis

SonarSource is a Swiss company founded in 2008. The company was born out of a desire to tackle and resolve the growing issues related to software quality, and to bring to market a solution that could track code quality in the software development process. In a market with competitive offerings from a handful of providers, SonarSource's portfolio is differentiated by its roots in open source, its accessibility, and a range of engaging options from a packaging and pricing perspective.

The SonarQube™ platform was created and taken to market (initially as "Sonar"), with SonarSource releasing its first commercial plug-in for the platform in October 2009. By March 2010, SonarSource began to see both community and enterprise acceptance of the SonarQube platform, which by then was being downloaded more than 2,000 times a month. In May 2010, SonarSource released COBOL and Visual Basic plug-ins, followed a few months later by a SQALE plug-in, the C# plug-in (June 2011), and the PL/SQL plug-in (September 2011).

The company's key intent is to bring affordable and intuitive quality solutions and analytics to development teams that also provide capabilities for broad, distributed use. Today, SonarSource has approximately 450 customers worldwide, including Deutsche Bank, Bank of America, Michelin, Telefónica, BNP Paribas, Thales, and EADS. The SonarQube platform is used by 30,000 to 40,000 organizations. The company has seen explosive growth over the past couple of years, and it now employs 35 staff, up from 20 people just 12 months ago.

### SonarQube Platform

SonarQube is a continuous quality analysis platform running as a web server sitting on a database that tracks metrics by analyzing code and code structure. SonarQube is an open source platform and was developed with a main objective in mind: "to make code quality management accessible to everyone with minimal effort." The SonarQube ecosystem is made up of the SonarQube Platform and a suite of plug-ins hosted on public infrastructure.

SonarQube essentially provides code analyzers, dashboards, reporting tools, issues tracking, and TimeMachine as core functionality, but it also has a plug-in mechanism enabling the community to extend the functionality (currently there are more than 60 plug-ins available).

SonarQube can become a coordinating hub for source code quality as it does not need to be restricted to developers or the technically savvy but can provide helpful information broadly to project managers, technical leads, IT, and even business leadership within an organization with customizable dashboards. SonarQube's architecture and plug-ins (such as SQALE) and the opportunity to manage and track technical debt can provide key information to managers and the business to proactively address defects iteratively throughout the software life cycle. They provide a high-level overview of the project standing as it relates to quality and cost, and to help address risk.

SonarQube provides more than mere high-level indicators about software health. As developers are provided with information at a granular code level, SonarQube enables those building software to find and to drill down to where code problems exist. SonarSource's product portfolio enables feedback and impact analysis on areas of software change, and provides feedback on how to improve development approaches. The products also provide coordination with build management to help enable continuous integration for deployment.

SonarQube integrates with tools like FindBugs, Checkstyle, PMD, FXCop, and CppCheck out-of-the-box or with provided plug-ins. It then can act as a central hub for code analysis tools, thereby providing historical insight and trend analysis for multiple projects. The all-in-one-place analytics and reports are a plus even if organizations choose not to act on the reports that are sent. Having access to historical data about code complexity and the number of issues tools spotted in the code can provide visibility into whether software is being built effectively or if the approach needs to be changed.

In terms of programming languages, SonarQube supports analysis of more than 20 languages such as Java, COBOL, C++, PL/SQL, and C# through plug-ins (open source or commercial) as the reporting engine is language agnostic.

SonarQube enables organizations to cover quality on seven axes and to report on:

- Duplicated code
- Coding standards
- Unit tests
- Complex code
- Potential bugs
- Comments
- Design and architecture

The most current version has also improved the evaluation of software quality attributes and does a better job of scoping technical debt and isolating the problems that create technical debt, according to references. SonarQube is configurable and can give a "grade" to the code (e.g., A, B, C, D) and can identify what it would typically cost in terms of effort and the type of effort needed to improve the software. For instance, the code might have a low rating due to lack of unit testing, or due to high amounts of duplicate code or security violations. SonarQube will show what the violations are and will estimate costs to address them. Users can change the effort estimations and they'll be calculated in, which is helpful. This release can make comparative team assessments across projects and parts of the organization as needed.

While SonarQube can be used tactically for one-off audits, it can be leveraged more strategically as a shared, common source of information for quality analysis as was just described, to help support a continuous improvement strategy for code quality.

A wide variety of organizations use SonarQube given the range of offerings of the portfolio (from the company's free "open source" option to the enterprise, site-wide "Ultimate" license version). However, a target at the high end is large and very large corporations with enterprise, distributed development teams and partner coordination. One-person teams can of course use the open source version, which can act as an on-ramp to adoption. But once an organization has crossed the line in terms of the number of development projects and users there is a need to move to the commercial enterprise solution. IDC spoke with four enterprise customer references for SonarQube, most of which began with open source adoption to gain a foothold and then evolving through to deployments of 1,000+ users.

### *Typical Size of Implementations and Benefits*

Large internationally operating organizations can be running analysis of more than 10,000 projects and analyzing 650-700 million lines of code in 14 languages with 8,000 visits per day on the website.

One of the customers IDC spoke with is tracking 1,200 projects with 160 million lines of code being scanned through SonarQube, plus a further 300 projects with another 160 million lines of code being scanned. Another customer ramped up from two dozen projects to over 2,230 projects for registered users with many more browsing dashboards anonymously.

## Why SonarQube?

The SonarQube references IDC spoke with needed a way to measure and enforce software and code quality metrics. A key goal was to have quantitative measurements of code quality and analyze those metrics to come up with a set of benchmark measurements – essentially, to utilize the platform to encourage good practices (and to discourage bad ones).

When evaluating competitive products they looked for the following: the features for quality analysis provided (such as dead code analysis, impact analysis, cross platform analysis), languages supported (SonarSource supports 20+), the flexibility of code review, and dashboard offerings and reporting analytics. Service organizations also evaluated the tools based on commercial and engagement constraints.

The advantages of SonarQube typically include its overall ease of use, requiring less time to learn and to adopt. Packaging options with SonarQube were also a benefit for both end users and service providers – enterprise licenses with "no strings attached" are a help to end users with dynamic distribution needs and for service providers, offering the freedom to be able to leverage SonarQube flexibly as part of engagements.

SonarQube is still evolving its support for impact analysis, though in the meantime some customer references have created workarounds to address the issue.

## The Benefits of Using SonarQube

The functionality that SonarQube customers described as being most significant in solving their core problems included the following:

- Code and quality visibility to be able to see where the hot spots are in an application to proactively include app quality "up front" as an initial and iterative part of the development process.
- Dashboards where users can pick and choose the metrics to contextualize and customize reporting.
- The ability to consolidate metrics at varying levels with different views – at the customer level, at the developer level, and/or at the business unit level – and to roll them up into "one source of truth" for a single portal or single point where everyone can go and see what they need to know.
- Managers/directors can customize and use SonarQube to measure the performance of individual groups – service providers can customize the dashboards for each of the customer organizations where they are working to address different kinds of needs and standards. They can also augment existing rules and integrate the results because SonarQube gives that level of flexibility.
- At the same time, organizations must be careful not to use SonarQube as a "bludgeon" to force "good behavior" – successful companies have leveraged the information to encourage better practices rather than to establish a "wall of shame" to punish individuals for poor coding behavior. This means using SonarQube as "diagnostic metrics" rather than "outcome based" metrics to better drive success.
- Overall, these capabilities have enabled customers to manage and alleviate technical debt through a cost-effective solution that can scale to an enterprise level and be broadly distributed. SonarQube helps organizations to benchmark code quality and to understand

how well their organizations are doing and how they can and have improved over time via information that is both qualitative and quantitative.

For application services organizations and application outsourcers specifically, SonarQube can help to ensure customers that the service provider is doing an effective job due to the visibility into code quality (as well as base lining and benchmarking code creation activities). Service providers can:

- **Use SonarQube internally to build higher levels of customer confidence.** New product releases are more stable, which enhances trust and confidence in the development team. As one service provider said: "SonarQube helps us analyze the situation, take actions, and quantify the improvement." Service providers are also using it to help prove continuous improvement. When installed at a customer site, SonarQube can help provide tangible evidence to the customer that the service provider is doing a good job and meeting SLAs.
- **Reduce business risk and total cost of ownership.** SonarQube can be used within application maintenance engagements to demonstrate the quality of the code that is being fixed, to help reduce both risk and total cost of ownership for customers.
- **Use SonarQube as an enterprisewide tool across engagements at all project levels to help determine code quality.** By provisioning a central tool that can instantly evaluate and monitor the standings of any project with respect to defined quality benchmarks, services organizations identified SonarQube as a scalable, cost-effective solution. One organization, for example, is now simulating and running 1,200 projects across both application development and application maintenance through SonarQube. This currently represents 20% of its clients and it continues to scale. At the same time, it has installed and is running SonarQube at a further 6% of clients, representing an additional 350 individual projects.
- **Ensure continuity in higher quality software development.** While it can take time for teams to understand how to use it and to improve the quality of the project, usage breeds adoption. Mandating adoption across an enterprisewide program allows providers to check the quality of code and make that visible to customers. Code profiling is key in this context.
- **Leverage SonarQube as a non-competitive or non-threatening partner.** Ecosystem sourcing is becoming the new normal, and service providers are increasingly looking at how to successfully move from being custodians of data and process to playing the role of guardian within this ecosystem. Additionally, we are in a period of "land grab" in which service providers' alliances and partnerships are simultaneously more important to customers but more difficult to evaluate, given the number of relationships different companies have. Leveraging companies such as SonarQube can help to support end-user clients with differentiated, targeted service offerings.

## Broader Implications and Opportunities with Use of SonarQube

Beyond scanning, an interesting outcome seen at some organizations is that individuals have started to act on the information made available by SonarQube to change and improve their behavior with regards to quality code creation. A number of teams have become engaged – and even excited – about checking factors such as test code coverage and have been able to improve it dramatically. Directors have used SonarQube information about rules violations to educate teams about shifting poor habits to improve code creation. A key outcome of SonarQube use at those organizations has been the opportunity to help encourage and even drive the right behavior.

Some organizations have helped this along by mandating the integration of metrics with the build and release process. This means that certain standards have to be met to allow the build and release process to move forward. While you can't force people to look at poor outcomes generally, when project deadlines and software release dates slip, there is an opportunity to work with engineering managers with specific data points and benchmarking to understand the impact of

prior work. (The organizations doing this are judicious in the standards that they put in place for stopping the build and release process.)

Overall, a tangible benefit has been the doubling of test coverage seen by some of the testing component teams using SonarQube. Some organizations have observed test team coverage increases of 4-5 times for unit tests, as well as an improvement in the thoroughness of those completed unit tests with use of SonarQube. This, in turn, helps drive accurate benchmarks for team progress along with the delivery of higher quality code.

In short, SonarQube can provide a single point for code analysis where users from developers to managers can go and see what they need to know to help improve code quality, and potentially to integrate with code review tools to include code review metrics in the dashboard.

## The Challenges for SonarQube

One of the key challenges for SonarSource is convincing organizations about the ROI benefits of implementing and using a code analysis solution. Part of the issue in this context is that it's challenging for organizations to comprehend long-term benefits at a time when many companies are seeking quick, iterative deployments. So it is important to understand the long-term benefits across the lifetime of the code, rather than merely the short-range advantages (which are significant in their own way but are not strategic). The evolution of the product to address functional capabilities such as impact analysis and cross-platform analysis are also areas where SonarSource needs to focus, according to customers. Integration with code review tools was also mentioned as an area of future focus.

SonarSource's small size can be a barrier to adoption for enterprise deployment decisions. But the fact that the base product is open source has typically jump started adoption for initial usage, and provides some reassurance for enterprise use of the commercial product as well. The enterprise customers that have demonstrated adoption of SonarQube with bigger deployments also serve to reassure those with concerns about the ability of the product portfolio to be adopted broadly.

## CONCLUSION: EVOLVING AND IMPROVING CODE QUALITY FOR BUSINESS AND CLIENT VALUE ENHANCEMENT

---

System integrators, outsourcers, and application service providers are continuously challenged in how to provide the next level of value to their clients at the same time as further improving internal processes to deliver better quality and lower-cost services. Quality assurance in terms of the services, solutions, and software delivered is a critical focus area, and the key is looking for and utilizing tools that can provide significant value to clients and help steer them toward success, as well as aiding in internal process improvement.

Clients are looking for more trusted relationships and more "skin in the game" from their providers. Just as consumers expect personalization, so do enterprises – and this expectation will only strengthen in future. Organizations are realizing that visibility into software and code quality metrics can provide a valuable tool through which to curb risk, avoid and reduce unnecessary cost, and enhance trust in software delivery.

Software drives competitive advantage and corporate success now more than ever. As companies increasingly need to improve quality and as the consequences of poor approaches to software development are visible and can be deeply damaging to revenue generation and to customer and prospect engagement, we see an urgent need for improvements in behavior with regards to code quality. Just as the "unexamined life is not worth living" so is unexamined code not worth

deploying. Neither companies nor the customers and prospects they are seeking to engage can afford to continue ineffective quality approaches.

A single portal such as that provided by SonarSource with SonarQube – with the ability to automate data gathering – is not just about quality per se but it is about enabling more thorough testing. Visibility into code quality provides the basis for effective decision making. Products like SonarQube can help organizations examine and understand software development through a single hub to start managing software development as a business.

Tracking metrics in software development to ensure continuity in quality software production brings several advantages, particularly to system integrators and service providers involved in multiple large-scale application development and implementation projects. Enhanced value is not only brought to the internal development team through process improvement and control, but to the client in terms of customer satisfaction – helping to build trust with the service provider.

As stated, differentiated services that deliver business value are key to winning future engagements. Service providers must position themselves and demonstrate that they are the essential "quality gate" between clients' IT and business organizations. In other words they are the essential layer that sits between these two functions charged with weeding out the rubbish before the business gets involved (again). This will allow the business to have more confidence in the applications that IT is pushing out, resulting in higher business satisfaction and helping IT position itself as a business value enabler versus cost center. Higher customer satisfaction or business satisfaction will not be derived from the use of SonarQube alone, however, and we recognize that SonarQube is one part of the "jigsaw" to achieving and managing client and business satisfaction.

Exploring ways to measure and enforce software and code quality metrics is increasingly necessary. Services organizations must put standards in place that can be adhered to and tracked across the entire business and across multiple projects.

Confidence in software delivery, a reduction in technical debt, a clear view of the pipeline and the software delivery process enabled by visibility and transparency into the code that's been created – all of these factors enable a collaborative working relationship between service providers and customers. Clients want service providers with which they can work closely and that have the appropriate domain expertise to execute well in these complex environments to ultimately enable innovation.

There are many aspects of "code quality" but we find that code analysis can be an essential building block in the pyramid of tools that helps improve the quality of code. We do caution, however, that enterprise adoption may not happen overnight, and that it will require commitment, dedication, and a company mandate to push adoption. Once it starts to scale, however, we find that usage breeds acceptance.

## About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

## Global Headquarters

5 Speen Street  
Framingham, MA 01701  
USA  
508.872.8200  
Twitter: @IDC  
idc-insights-community.com  
www.idc.com

---

### Copyright Notice

This IDC research document was published as part of an IDC continuous intelligence service, providing written research, analyst interactions, telebriefings, and conferences. Visit [www.idc.com](http://www.idc.com) to learn more about IDC subscription and consulting services. To view a list of IDC offices worldwide, visit [www.idc.com/offices](http://www.idc.com/offices). Please contact the IDC Hotline at 800.343.4952, ext. 7988 (or +1.508.988.7988) or [sales@idc.com](mailto:sales@idc.com) for information on applying the price of this document toward the purchase of an IDC service or for information on additional copies or Web rights.

Copyright 2015 IDC. Reproduction is forbidden unless authorized. All rights reserved.

